US006154571A

United States Patent [19]

Cox et al.

[11] Patent Number: 6,154,571

[45] Date of Patent: Nov. 28, 2000

[54] **ROBUST DIGITAL WATERMARKING**

[75] Inventors: **Ingemar J. Cox**, Lawrenceville; **Matthew L. Miller**, Princeton, both of N.J.; **Ryoma Oami**, Tokyo, Japan

[73] Assignees: **NEC Research Institute, Inc.**, Princeton, N.J.; **NEC Corporation**, Tokyo, Japan

[21] Appl. No.: **09/118,467**

[22] Filed: **Jul. 17, 1998**

Related U.S. Application Data

[60] Provisional application No. 60/090,532, Jun. 24, 1998.

[51] **Int. Cl.**^{7} **G06K 9/00**

[52] **U.S. Cl.** **382/250**; 382/232; 382/100

[58] **Field of Search** 382/100, 232, 382/250, 276, 251, 252, 244

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,939,515	7/1990	Adelson	341/51
5,319,735	6/1994	Preuss et al.	395/2.14
5,530,751	6/1996	Morris	380/4
5,530,759	6/1996	Braudaway et al.	380/54
5,568,570	10/1996	Rabbani	382/238
5,613,004	3/1997	Cooperman et al.	380/28
5,636,292	6/1997	Rhoads	382/232
5,646,997	7/1997	Barton	380/23
5,659,726	8/1997	Sandford, II et al.	395/612
5,664,018	9/1997	Leighton	380/54
5,687,236	11/1997	Moskowitz et al.	380/28
5,809,139	9/1998	Girod et al.	380/202
6,037,984	3/2000	Isnardi et al.	348/403

FOREIGN PATENT DOCUMENTS

0690595	1/1995	European Pat. Off.	H04L 9/18
2196167	4/1988	United Kingdom	G11B 20/08
8908915	9/1989	WIPO	G11B 21/10
9520291	7/1995	WIPO	H04N 1/32
9621290	7/1996	WIPO	H04H 1/00
9625005	8/1996	WIPO	H04N 7/08
9627259	9/1996	WIPO	H04N 1/32

OTHER PUBLICATIONS

R.G. Van Schyndel et al, "A digital watermark," in Intl. Conf. On Image Processing, vol. 2, pp. 86–90, 1994.

G. Caronni, "Assuring Ownership Rights for Digital Images," in Proc. Reliable IT Systems, VIS '95, 1995.

J. Brassil et al, "Electronic Marking and Identification Techniques to Discourage Document Copying," in Proc. Infocom '94, pp. 1278–1287, 1994.

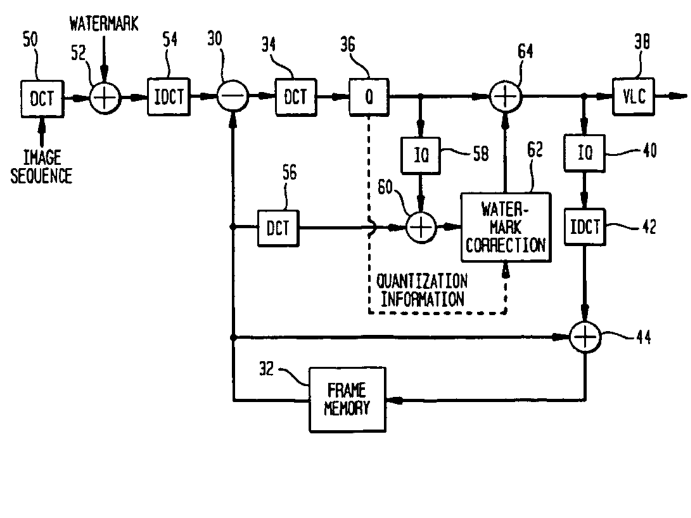
(List continued on next page.)

Primary Examiner—Bijan Tadayon
Assistant Examiner—Shervin Nakhjavan
Attorney, Agent, or Firm—Philip J. Feig

[57] **ABSTRACT**

A watermarking procedure that is applicable to images, audio, video and multimedia data to be watermarked divides the data to be watermarked into a set of n×n blocks, such as the 8×8 blocks of MPEG. The same watermark signal can be distributed throughout the set of blocks in a large variety of ways. This allows the insertion algorithm to be changed without affecting the decoders. The decoding procedure first sums together the DCT coefficients of N sets of 8×8 blocks to form a set of N summed 8×8 blocks and then extracts the watermark from the summed block. Since the sum of the DCT blocks is equal to the DCT of the sum of the intensity blocks, efficient decoding can occur in both the spatial and frequency domains. The symmetric nature of the decoding process allows geometric distortions to be handled in the spatial domain and other signal distortions to be handled in the frequency domain. Moreover, insertion of a watermark signal into image data and the subsequent extraction of the watermark from watermarked image data which has been subject to distortion between the times of insertion and extraction involves the insertion of multiple watermarks designed to survive predefined distortions of the image data, such as panscan or letterbox mode transformations. Alternatively, a registration pattern in the image data, after the image data containing the registration pattern is subject to an unknown distortion, is used to compensate for distortion of the watermarked image data.

10 Claims, 10 Drawing Sheets



OTHER PUBLICATIONS

K. Tanaka et al, "Embedding Secret Information into a Dithered Multi-level Image," in IEEE Military Comm. Conf., pp. 216–220, 1990.

K. Matsui et al, "Video-Steganography: How to Secretly Embed a Signature in a Picture," in IMA Intellectual Property Project Proc., vol. 1, pp. 187–206, 1994.

Macq and Quisquater, "Cryptology for Digital TV Broadcasting," in Proc. of the IEEE, vol. 83, No. 6, pp. 944–957, 1995.

W. Bender et al, "Techniques for data hiding," in Proc. of SPIE, vol. 2420, No. 40, Feb. 1995.

Koch, Rindfrey and Zhao, "Copyright Protection for Multimedia Data," in Proc. of the Int'l Conf. on Digital Media and Electronic Publishing (Leeds, UK, Dec., 6–8, 1994).

Koch and Zhao, "Towards Robust and Hidden Image Copyright Labeling," in Proc. of 1995 IEEE Workshop on Nonlinear Signal and Image Processing (Neos Marmaras, Halkidiki, Greece, Jun. 20–22, 1995.

"Digital Copyright: Who Owns What?" NewMedia, Sep. 1995, pp. 38–43.

"Publish and Be Robbed?" New Scientist, Feb. 18, 1995, pp. 32–37.

Kohno et al, "Spread Spectrum Access Methods for Wireless Communications," in IEEE Communications Magazine, Jan. 1995, pp. 58–67.

Campana and Quinn, "Spread spectrum communications," in IEEE Potentials, Apr. 1993, pp. 13–16.

Mowbray and Grant, "Wideband coding for uncoordinated multiple access communication," in Electronics & Communication Engineering Journal, Dec. 1992, pp. 351–361.

Digimarc Overview & "Wired" Magazine article (Jul. 1995 issue)—Jun. 1995.

A.G. Bors et al., "Image Watermarking Using DCT Domain Constraints", Dept. Of Informatics, University of Thessaloniki.

I.J. Cox et al., "Secure Spread Spectrum Watermarking for Multimedia", NEC Research Institute, Technical Report 95–10.

H.S. Stone, "Analysis of Attacks on Image Watermarks with Randomized Coefficients", NEC Research Institute, May 17, 1996.

F.M. Boland et al., "Watermarking Digital Images for Copyright Protection", Image Processing and its Applications, Jul. 4–6, 1995, Conference Publication No. 410, pp. 326–330.

L. Boney et al., "Digital Watermarks for Audio Signals".

Swanson et al., "Transparent Robust Image Watermarking", Proc. IEEE Int. Conf. On Image Proc. 1996.

J.J.K. O Ruanaidh et al., "Phase Watermarking of Digital Images".

I. Pitas, "A Method for Signature Casting on Digital Images".

C.T. Hsu et al., "Hidden Signatures in Images", ICIP 96 Conf. Proc., Sep. 16–19, 1996.

M. Schneider et al., "A Robust Content Based Digital Signature for Image Authentication", ICIP 96 Conf. Proc., Sep. 16–19, 1996.

S. Roche et al., "Multi-Resolution Access Control Algorithm Based on Fractal Coding", ICIP 96 Conf. Proc., Sep. 16–19, 1996.

K. Hirotsugu, "An Image Digital Signature System with ZKIP for the Graph Isomorphism", ICIP 96 Conf. Proc., Sep. 16–19, 1996.

R.B. Wolfgang et al., "A Watermark for Digital Images".

J.J.K. O Ruanaidh et al., "Watermarking Digital Images for Copyright Protection", EVA 96 Florence, pp. 1–7.

T. Aura, "Invisible Communication", Nov. 6, 1995.

D. Kahn, "Information Hiding—An Annotated Bibliography", Macmillan 1967, Library of Congress catalog No. 63–16109.

Craver et al., "Can Invisible Watermarks Resolve Rightful Ownership?", IBM Research Report, RC 20509, Jul. 25, 1996.

Podilchuk et al., "Digital Image Watermarking Using Visual Models", Proc. of EI'97, vol. 3016, Feb. 9–14, 1997.

Cox et al., "A review of watermarking and the importance of perceptual modeling", Proc. of EI'97, vol. 3016, Feb. 9–14, 1997.

Watson, "DCT quantization matrices visually optimized for individual images", SPIE, vol. 1913, pp. 202–216.

Ahumada, Jr. et al., "Luminance-Model-Based DCT Quantization for Color Image Compression", SPIE, vol. 1666 (1992), pp. 365–374.

Hartung et al., "Digital Watermarking of Raw and Compressed Video", Systems for Video Communication, Oct. 1996, pp. 205–213.

FIG. 1

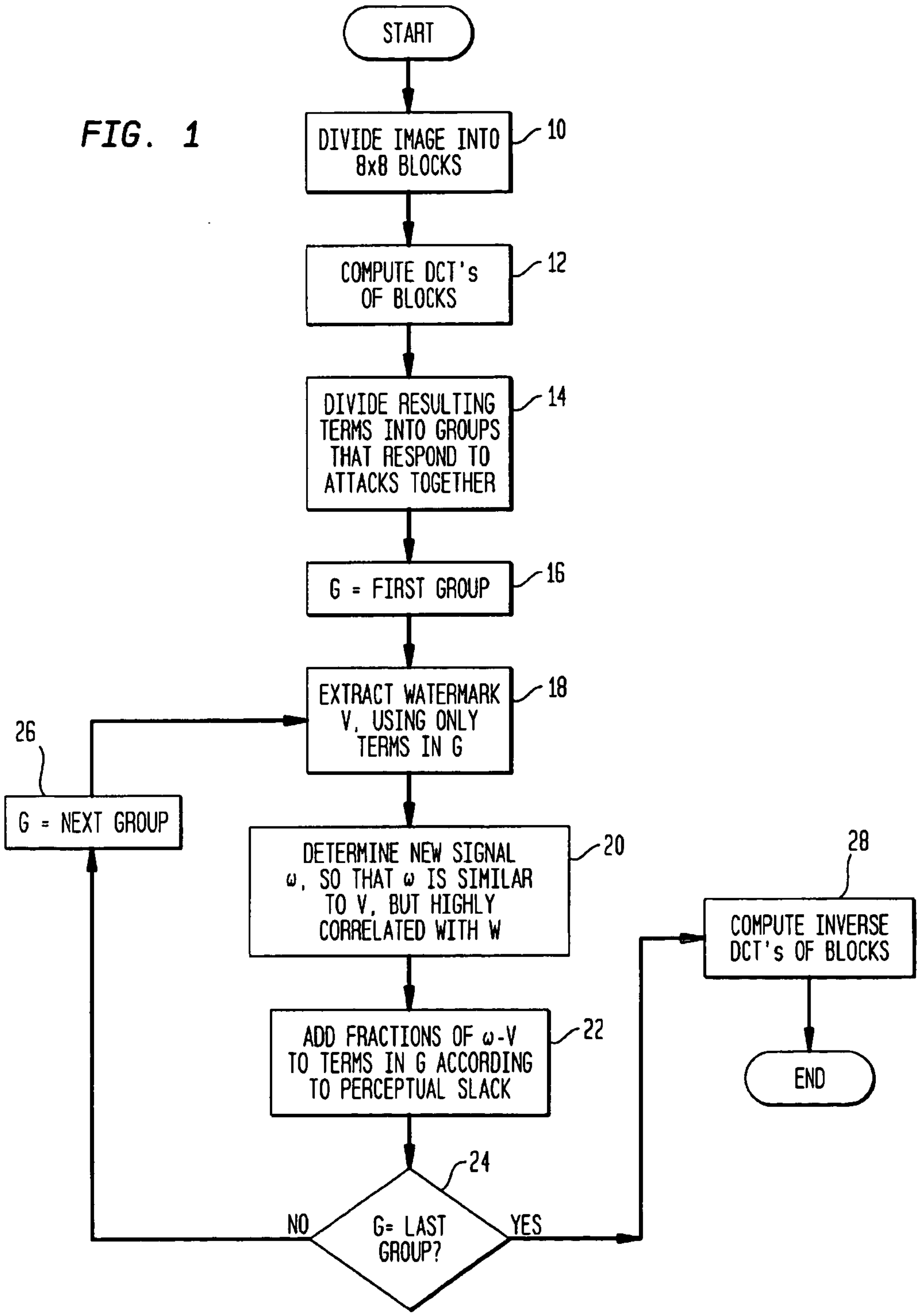

START
DIVIDE IMAGE INTO 8x8 BLOCKS
10
COMPUTE DCT's OF BLOCKS
12
DIVIDE RESULTING TERMS INTO GROUPS THAT RESPOND TO ATTACKS TOGETHER
14
G = FIRST GROUP
16
EXTRACT WATERMARK V, USING ONLY TERMS IN G
18
DETERMINE NEW SIGNAL ω, SO THAT ω IS SIMILAR TO V, BUT HIGHLY CORRELATED WITH W
20
ADD FRACTIONS OF ω-V TO TERMS IN G ACCORDING TO PERCEPTUAL SLACK
22
G= LAST GROUP?
24
NO
YES
G = NEXT GROUP
26
COMPUTE INVERSE DCT's OF BLOCKS
28
END

FIG. 2

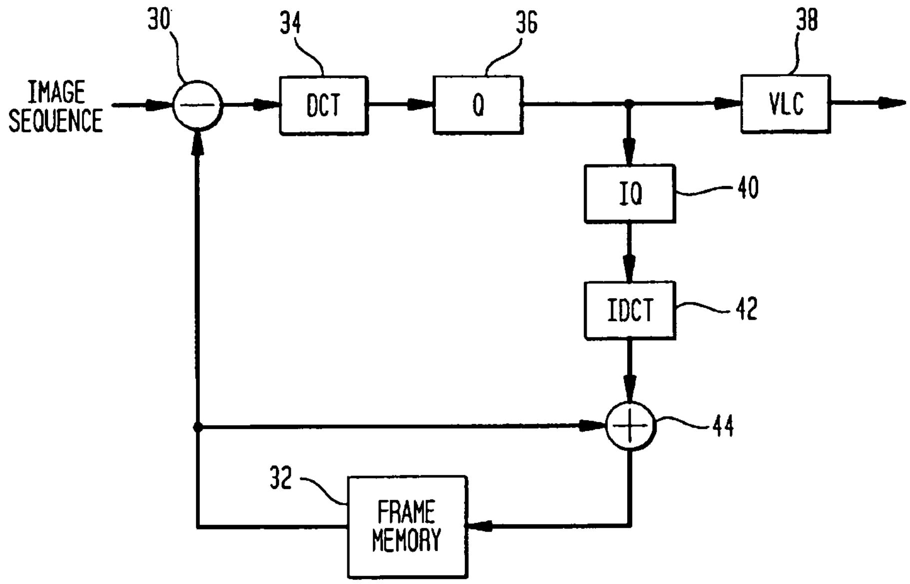

30
34
36
38
IMAGE SEQUENCE
DCT
Q
VLC
IQ
40
IDCT
42
44
32
FRAME MEMORY

FIG. 3

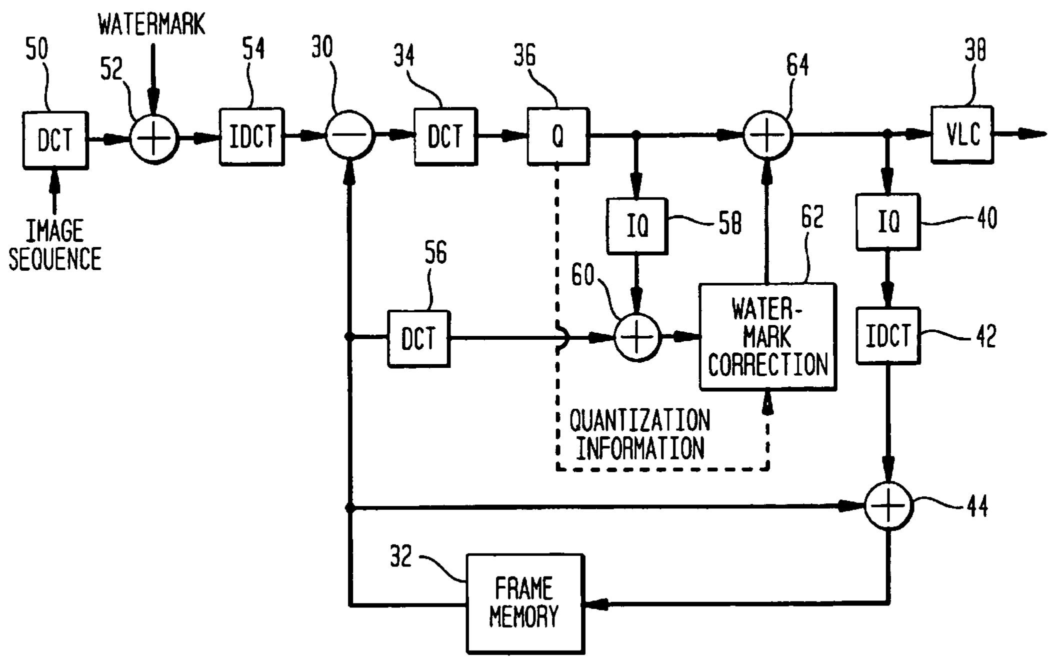

WATERMARK
50
52
54
30
34
36
64
38
DCT
IDCT
DCT
Q
VLC
IMAGE SEQUENCE
IQ
58
62
IQ
40
56
60
DCT
WATER-MARK CORRECTION
IDCT
42
QUANTIZATION INFORMATION
44
32
FRAME MEMORY

FIG. 4

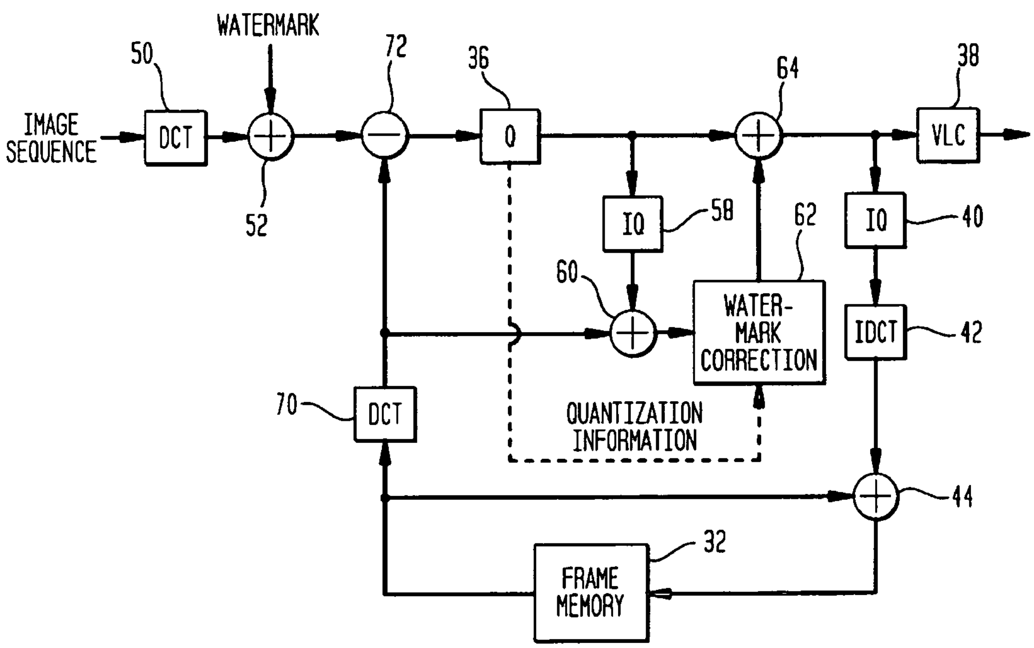

WATERMARK
50
72
36
64
38
IMAGE SEQUENCE
DCT
Q
VLC
52
IQ
58
62
IQ
40
60
WATER-MARK CORRECTION
IDCT
42
70
DCT
QUANTIZATION INFORMATION
44
32
FRAME MEMORY

FIG. 5

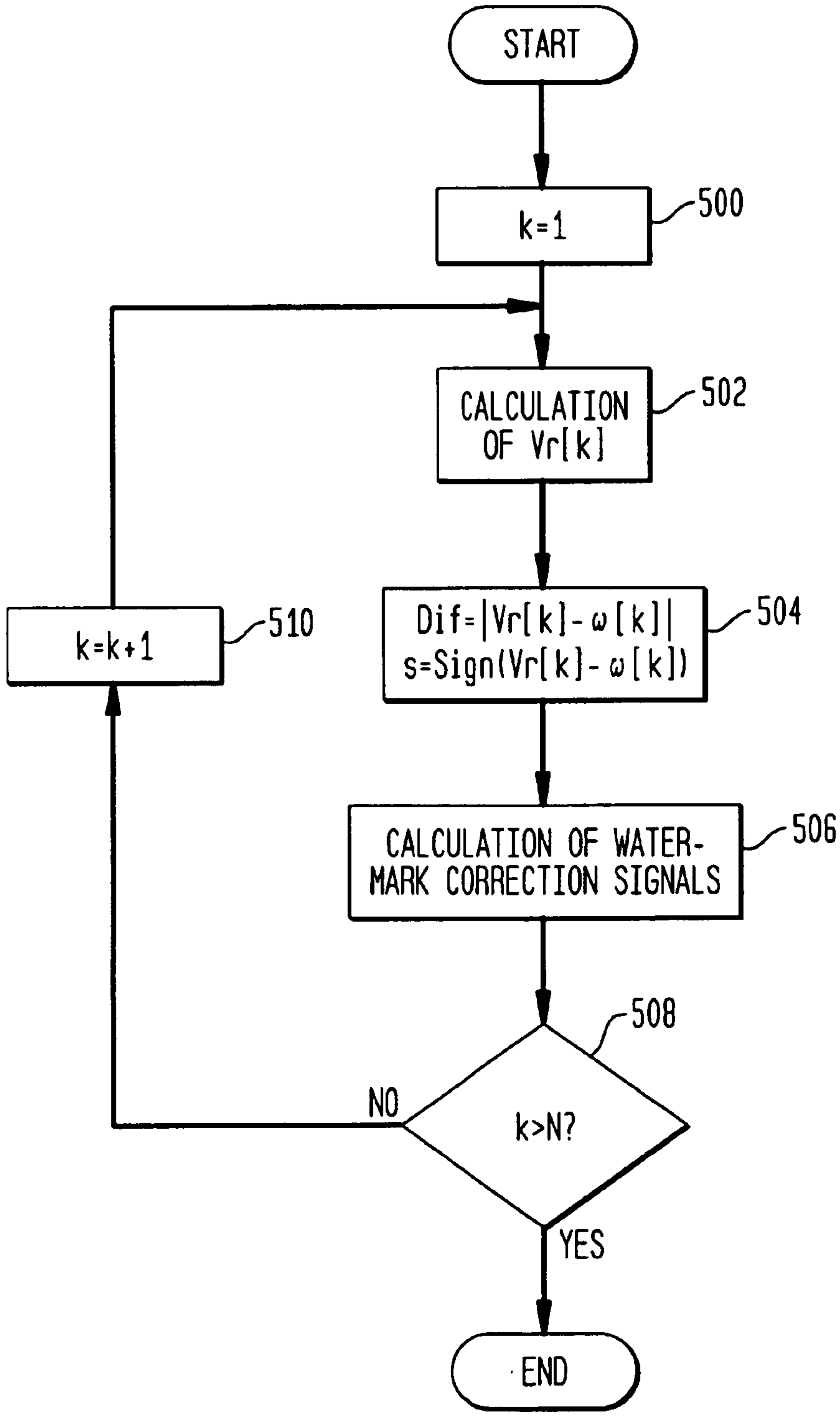

START
k=1
500
CALCULATION OF Vr[k]
502
Dif=|Vr[k]-ω[k]|
s=Sign(Vr[k]-ω[k])
504
CALCULATION OF WATER-MARK CORRECTION SIGNALS
506
k>N?
508
NO
YES
k=k+1
510
END

FIG. 6

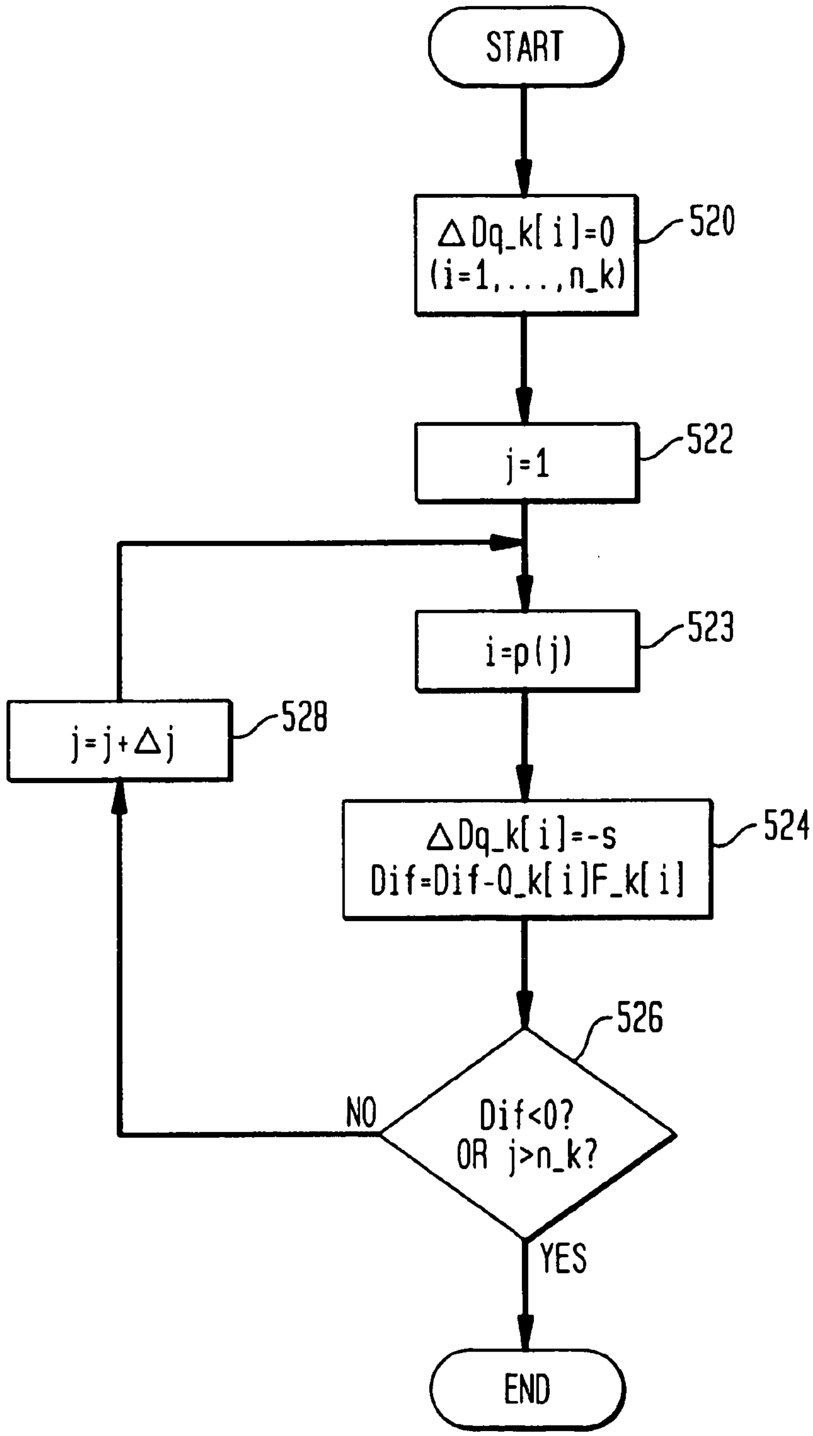

START
ΔDq_k[i]=0
(i=1,...,n_k)
520
j=1
522
i=p(j)
523
j=j+Δj
528
ΔDq_k[i]=-s
Dif=Dif-Q_k[i]F_k[i]
524
526
Dif<0?
OR j>n_k?
NO
YES
END

FIG. 7

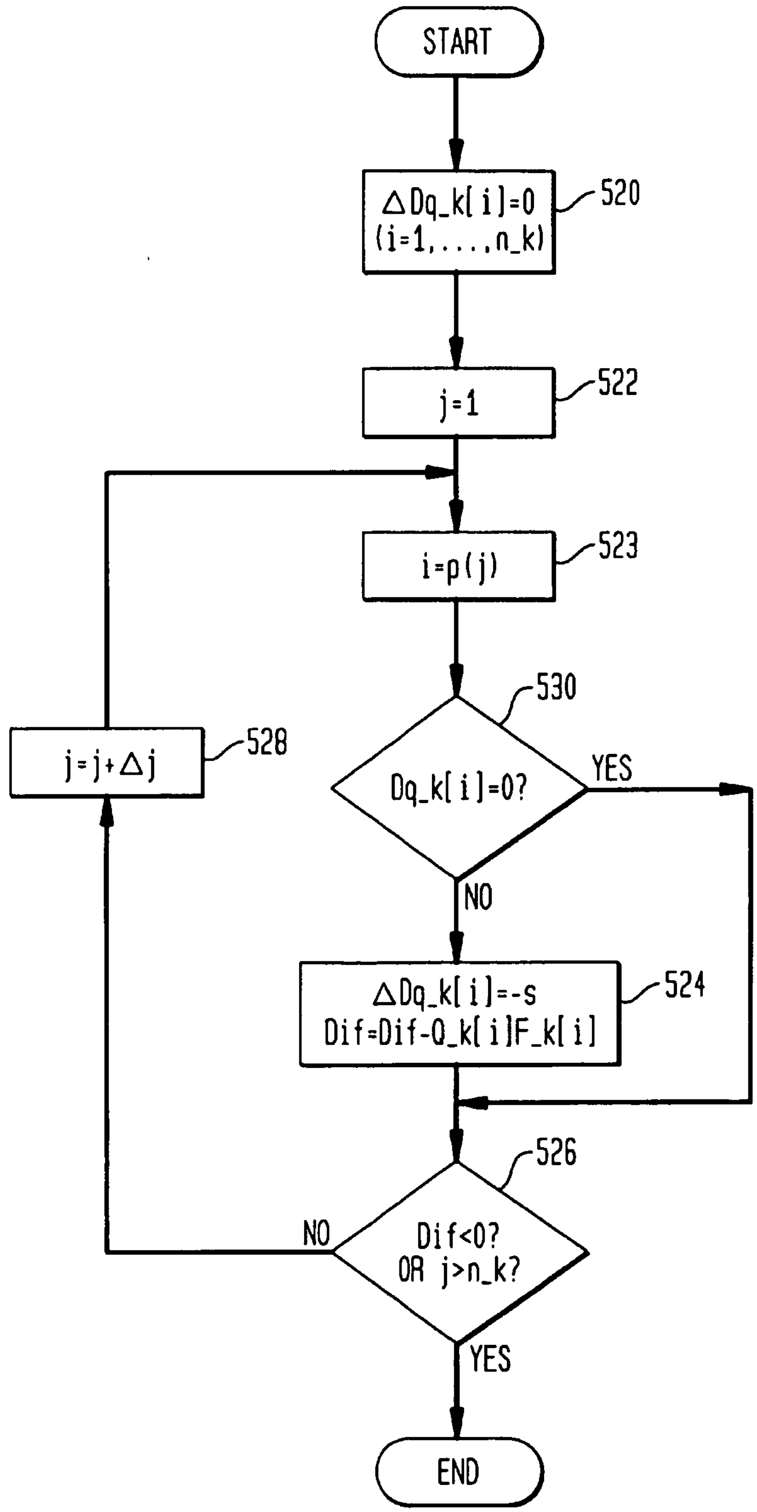
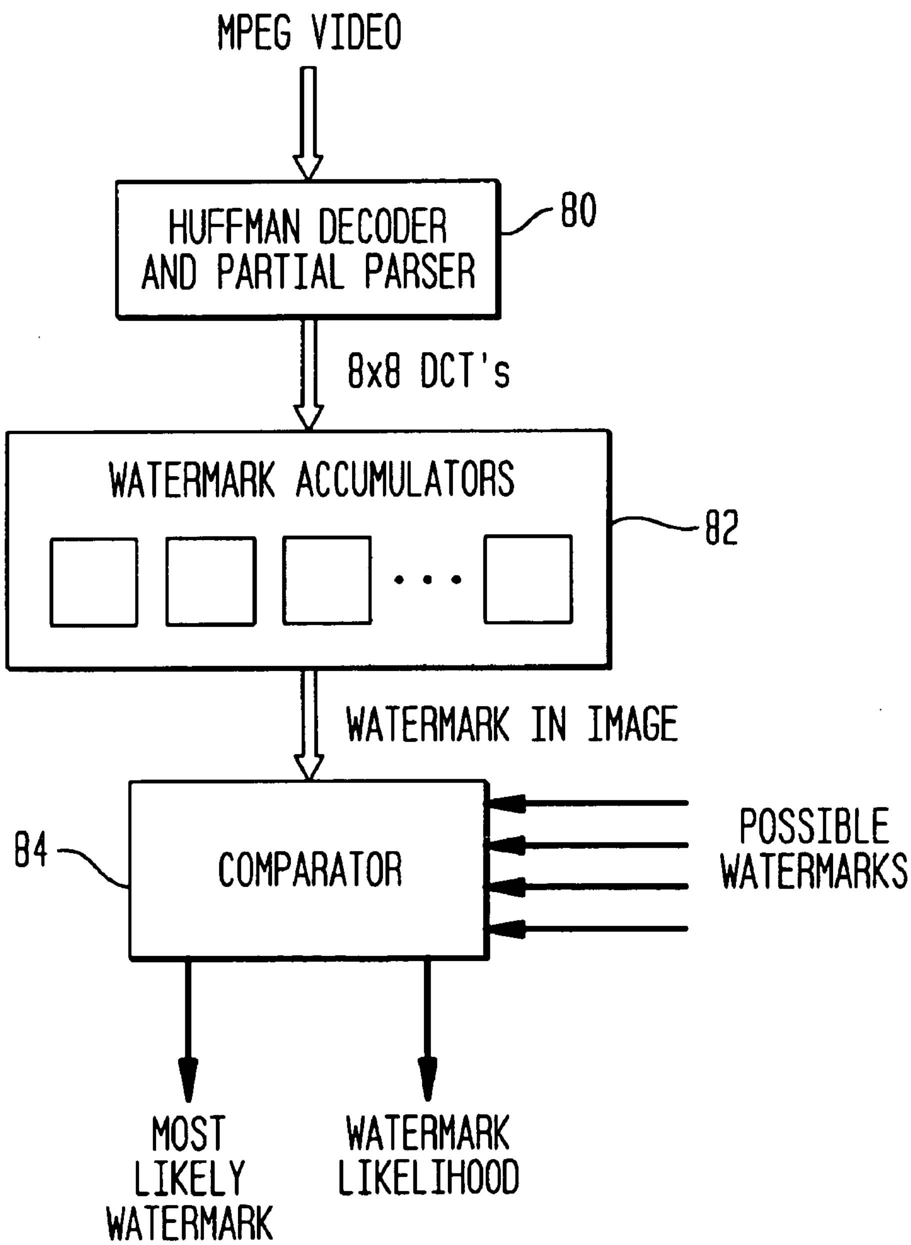

START
ΔDq_k[i]=0
(i=1,...,n_k)
520
j=1
522
i=p(j)
523
530
Dq_k[i]=0?
YES
NO
j=j+Δj
528
ΔDq_k[i]=-s
Dif=Dif-Q_k[i]F_k[i]
524
526
Dif<0?
OR j>n_k?
NO
YES
END

FIG. 8



MPEG VIDEO
HUFFMAN DECODER AND PARTIAL PARSER
80
8x8 DCT's
WATERMARK ACCUMULATORS
82
WATERMARK IN IMAGE
84
COMPARATOR
POSSIBLE WATERMARKS
MOST LIKELY WATERMARK
WATERMARK LIKELIHOOD

FIG. 9

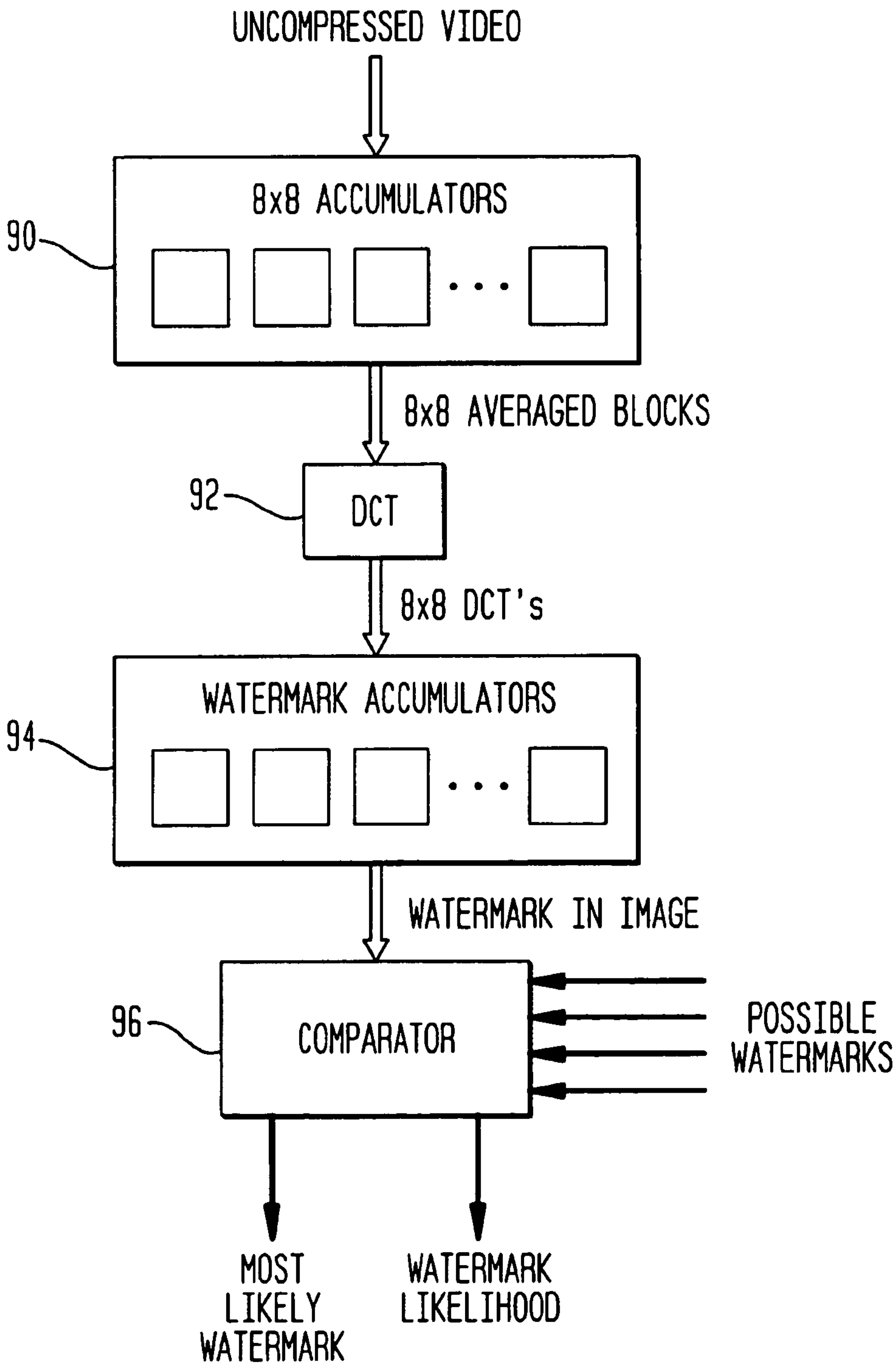

UNCOMPRESSED VIDEO
8x8 ACCUMULATORS
90
8x8 AVERAGED BLOCKS
92
DCT
8x8 DCT's
WATERMARK ACCUMULATORS
94
WATERMARK IN IMAGE
96
COMPARATOR
POSSIBLE WATERMARKS
MOST LIKELY WATERMARK
WATERMARK LIKELIHOOD

FIG. 10

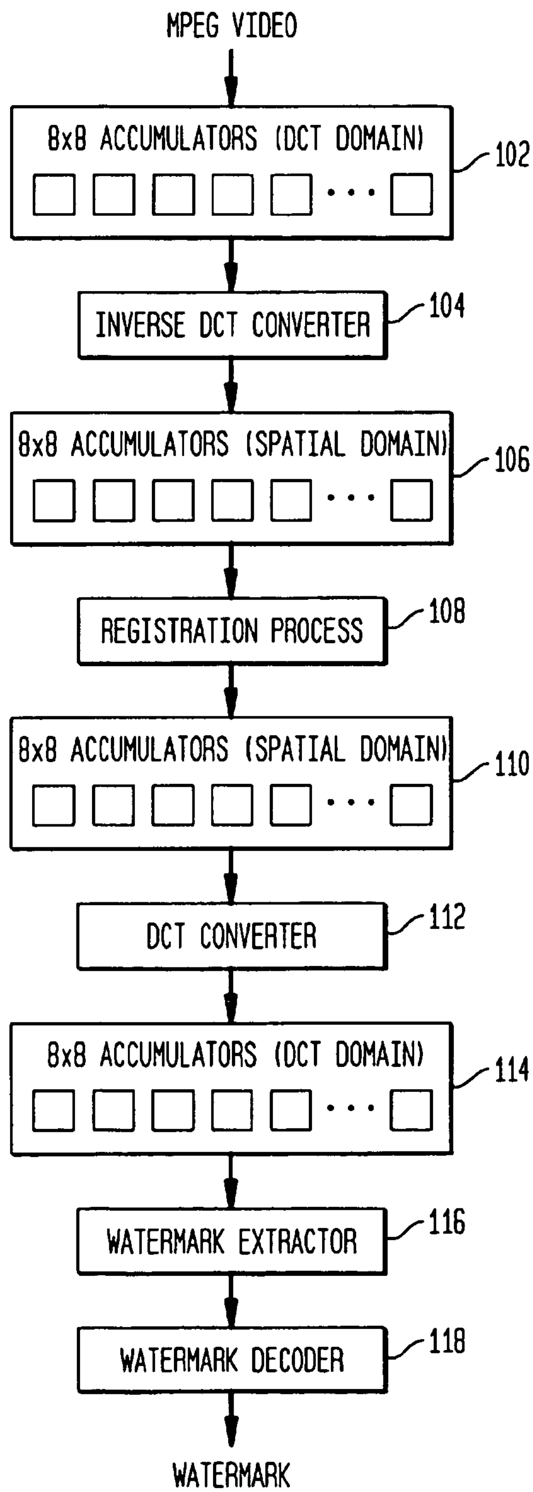
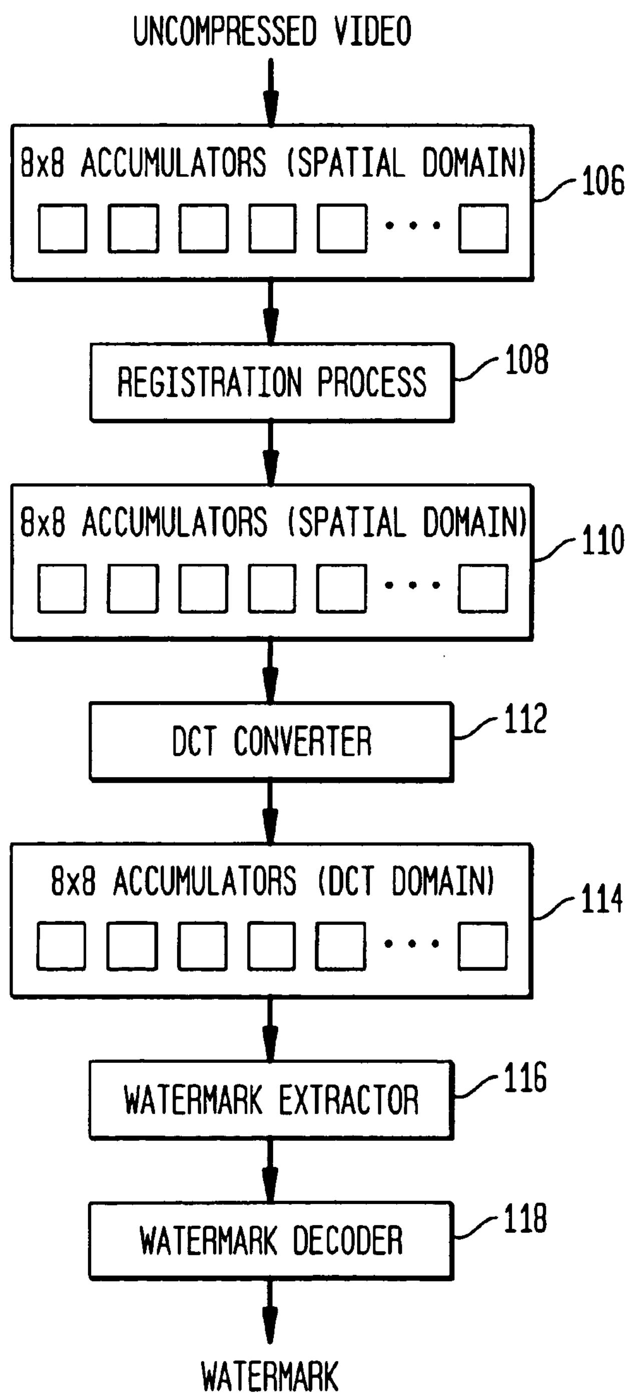

MPEG VIDEO
8x8 ACCUMULATORS (DCT DOMAIN)
102
INVERSE DCT CONVERTER
104
8x8 ACCUMULATORS (SPATIAL DOMAIN)
106
REGISTRATION PROCESS
108
8x8 ACCUMULATORS (SPATIAL DOMAIN)
110
DCT CONVERTER
112
8x8 ACCUMULATORS (DCT DOMAIN)
114
WATERMARK EXTRACTOR
116
WATERMARK DECODER
118
WATERMARK

FIG. 11


UNCOMPRESSED VIDEO
8x8 ACCUMULATORS (SPATIAL DOMAIN)
106
REGISTRATION PROCESS
108
8x8 ACCUMULATORS (SPATIAL DOMAIN)
110
DCT CONVERTER
112
8x8 ACCUMULATORS (DCT DOMAIN)
114
WATERMARK EXTRACTOR
116
WATERMARK DECODER
118
WATERMARK

ROBUST DIGITAL WATERMARKING

CROSS-REFERENCE TO RELATED APPLICATIONS

This is a conversion of provisional application Ser. No. 60/090,532, filed Jun. 24, 1998.

FIELD OF THE INVENTION

The present invention relates to digital watermarking of data including image, video and multimedia data. Specifically, the invention relates to insertion and detection or extraction of embedded signals for purposes of watermarking, in which the insertion and detection procedures are applied to sums of subregions of the data. When these subregions correspond to the 8×8 pixel blocks used for MPEG and JPEG compression and decompression, the watermarking procedure can be tightly coupled with these compression algorithms to achieve very significant savings in computation. The invention also relates to the insertion and detection of embedded signals for the purposes of watermarking, in which the watermarked data might have undergone distortion between the times of insertion and detection of the watermark.

BACKGROUND OF THE INVENTION

The proliferation of digitized media such as image, video and multimedia is creating a need for a security system that facilitates the identification of the source of the material.

Content providers, i.e. owners of works in digital data form, have a need to embed signals into video/image/multimedia data, which can subsequently be detected by software, and/or hardware devices for purposes of authentication of copyright ownership, and copy control and management.

For example, a coded signal might be inserted in data to indicate that the data should not be copied. The embedded signal should preserve the image fidelity, be robust to common signal transformations and resistant to tampering. In addition, consideration must be given to the data rate that can be provided by the system, though current requirements are relatively low—a few bits per frame.

In U.S. patent application Ser. No. 08/534,894, filed Sep. 28, 1995, entitled "Secure Spread Spectrum Watermarking for Multimedia Data", which is incorporated herein by reference, there was proposed a spread spectrum watermarking method which embedded a watermark signal into perceptually significant regions of an image for the purposes of identifying the content owner and/or possessor. A strength of this approach is that the watermark is very difficult to remove. In fact, this method only allows the watermark to be read if the original image or data is available for comparison. This is because the original spectrum of the watermark is shaped to that of the image through a non-linear multiplicative procedure, and this spectral shaping must be removed prior to detection by matched filtering. In addition, the watermark is usually inserted into the N largest spectral coefficients, the ranking of which is not preserved after watermarking. This method does not allow software and hardware devices to directly read embedded signals without access to the original unwatermarked material.

In an article by Cox et al., entitled "Secured Spectrum Watermarking for Multimedia" available at http://www.neci.nj.nec.con/tr/index.html (Technical Report No. 95-10) spread spectrum watermarking is described which embeds a pseudo-random noise sequence into the digital data for watermarking purposes.

The above prior art watermark extraction methodology requires the original image spectrum be subtracted from the watermark image spectrum. This restricts the use of the method when there is no original image or original image spectrum available to the decoder. One application where this presents a significant difficulty is for third party device providers desiring to read embedded information for operation or denying operation of such a device.

In U.S. Pat. No. 5,319,735 by R. D. Preuss et al entitled "Embedded Signaling" digital information is encoded to produce a sequence of code symbols. The sequence of code symbols is embedded in an audio signal by generating a corresponding sequence of spread spectrum code signals representing the sequence of code symbols. The frequency components of the code signal being essentially confined to a preselected signaling band lying within the bandwidth of the audio signal and successive segments of the code signal corresponds to successive code symbols in the sequence. The audio signal is continuously frequency analyzed over a frequency band encompassing the signaling band and the code signal is dynamically filtered as a function of the analysis to provide a modified code signal with frequency component levels which are, at each time instant, essentially a preselected proportion of the levels of the audio signal frequency components in corresponding frequency ranges. The modified code signal and the audio signal are combined to provide a composite audio signal in which the digital information is embedded. This component audio signal is then recorded on a recording medium or is otherwise subjected to a transmission channel. Two key elements of this process are the spectral shaping and spectral equalization that occur at the insertion and extraction stages, respectively, thereby allowing the embedded signal to be extracted without access to the unwatermarked original data.

In U.S. patent application Ser. No. 08/708,331, filed Sep. 4, 1996, entitled "A Spread Spectrum Watermark for Embedded Signaling" by Cox, and incorporated herein by reference, there is described a method for extracting a watermark of embedded data from watermarked images or video without using an original or unwatermarked version of the data.

This method of watermarking an image or image data for embedded signaling requires that the DCT (discrete cosine transform) and its inverse of the entire image be computed. There are fast algorithms for computing the DCT in N log N time, where N is the number of pixels in the image. However, for N=512×512, the computational requirement is still high, particularly if the encoding and extracting processes must occur at video rates, i.e. 30 frames per second. This method requires approximately 30 times the computation needed for MPEG-II decompression.

One possible way to achieve real-time video watermarking is to only watermark every Nth frame. However, content owners wish to protect each and every video frame. Moreover, if it is known which frames contain embedded signals, it is simple to remove those frames with no noticeable degradation in the video signal.

An alternative option is to insert the watermark into n×n blocks of the image (subimages) where n<<N. If the block size is chosen to be 8×8, i.e. the same size as that used for MPEG image compression, then it is possible to tightly couple the watermark insertion and extraction procedures to those of the MPEG compression and decompression algorithms. Considerable computational saving can then be achieved since the most expensive computations relate to the calculation of the DCT and its inverse and these steps are

already computed as part of the compression and decompression algorithm. The incremental cost of watermarking is then very small, typically less than five percent of the computational requirements associated with MPEG.

U.S. patent application Ser. No. 08/715,953, filed Sep. 19, 1996, entitled "Watermarking of Image Data Using MPEG/JPEG Coefficients" which is incorporated herein by reference, advances this work by using MPEG/JPEG coefficients to encode the image data.

U.S. patent application Ser. No. 08/746,022, filed Nov. 5, 1996, entitled "Digital Watermarking", which is incorporated herein by references, describes storing watermark information into subimages and extracting watermark information from subimages.

A review of watermarking is found in an article by Cox et al., entitled "A review of watermarking and the importance of perceptual modeling" in Proc. of EI'97, vol. 30–16, Feb. 9–14, 1997.

There have been several proposals to watermark MPEG video or JPEG compressed still images. In all cases, each 8×8 DCT block is modified to contain the watermark or a portion thereof. Consequently, decoding of the watermark requires that each 8×8 block be individually analyzed to extract the watermark signal contained therein. The individual extracted signals may then be combined to form a composite watermark, which is then compared with known watermarks. Because each block must be analyzed individually, an uncompressed image must be converted back to the block-based DCT representation, which is computationally expensive. Thus, while the decoder may be computationally efficient in the DCT domain, extracting a watermark from the spatial domain is much more expensive.

To allow for computationally efficient detection of the watermark in both the spatial and DCT domains, a watermark may be inserted in the sum of all the 8×8 blocks in the DCT domain, or the sum of a subset of all the 8×8 blocks in the DCT domain. A major advantage of this approach is that if the image is only available in the spatial domain, then the summation can also be performed in the spatial domain to compute a small set of summed 8×8 blocks and only those blocks must then be transformed into the DCT domain. This is because the sum of the DCT blocks is equal to the DCT of the sum of the intensities. Thus, the computational cost of decoding in the DCT and spatial domains is approximately the same.

A second advantage of watermarking the sum of the DCT blocks is that there are an unlimited number of equivalent methods to apportion the watermark throughout the image. For example, if the watermark requires a change of Δi to the i'th coefficient of the summed DCT block, then, if there are M blocks in the image, $\Delta i/M$ can be added to each individual block, or block 1 can have Δi added to it and the remaining M-1 blocks left unaltered, ignoring for the moment issues of image fidelity. Because of this one to many mapping, it is possible to alter the insertion algorithm without changing the decoder. This is a very important characteristic, since in some watermarking applications, there may be many hardware decoders that are deployed, such that changing the decoder is impractical. However, improvements to the insertion algorithm can still result in improved detection using the approach described herein.

A third advantage of watermarking the sum of the DCT blocks is that watermark signals extracted from these sums have small variances, compared with the amount that they may be changed without causing fidelity problems. This means that, in many cases, it is possible to change an image so that the summed DCT blocks perfectly match the required watermark signal, even though the resulting image appears identical to the original.

Finally, it is well known that some problems, such as modeling the human visual system, are best performed in the frequency domain, where other problems such as geometric transformations are more conveniently dealt with in the spatial domain. Since the computational cost of decoding the watermark is now symmetric, it is possible to switch from spatial to frequency domains at will in order to correct for various signal transformations that may corrupt the watermark.

SUMMARY OF THE INVENTION

The present invention concerns a novel insertion method which employs a specific model of the human visual system which provides much better control over image fidelity. Tests have shown that it is possible to obtain large signals (more than 15 standard deviations from 0 correlation) with images that are indistinguishable from their respective original images.

The method handles robustness against various types of attacks in ways that are easy to relate to the specific type of attack.

The method is adaptable so that the model of the human visual system and the techniques used for handling attacks can be changed later without having to change the detector. The result is that it is possible to continue improving watermarking, particularly DVD (digital video disk) watermarking, even after many detectors have been installed. This is analogous to the situation with MPEG video for which encoder technology can be improved without having to change existing decoders.

Use of the present insertion method allows a simple detection algorithm in either MPEG or decompressed domains.

The invention also concerns a novel detection method which is easy to implement, easy to analyze and has a low computational cost, whether the incoming video is MPEG compressed or uncompressed.

The present invention also concerns a novel insertion method that hides multiple patterns in the data. These patterns fall into two categories: 1) registration patterns used during detection to compensate for translational shifts, and 2) watermark patterns that encode the information content of the watermark.

A principal object of the present invention is the provision of a digital watermark insertion method which allows detection of watermarks after the watermarked data is subjected to predefined scale changes, without modification to the watermark detector.

Another object of the invention is the provision of a watermark detection method that is computationally inexpensive in either the MPEG or decompressed domains.

A still other object of the invention is the provision of a digital watermarking method that withstands attacks without having to change a detector.

Further and still other objects of the invention will become more clearly apparent when the following description is read in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a general data flow diagram of a method of inserting a watermark into media data;

FIG. 2 is a schematic diagram of an MPEG-2 encoder;

FIG. 3 is a schematic diagram of a modified MPEG-2 encoder for reducing degradation of a watermark in watermarked data;

FIG. 4 is a schematic diagram of an alternative modified MPEG-2 encoder for reducing degradation of a watermark in watermarked data;

FIG. 5 is a flow chart of the process performed in the watermark correction device in FIGS. 3 and 4;

FIG. 6 is a flow chart of the process performed in step 506 of FIG. 5;

FIG. 7 is a flow chart of an alternative process performed in step 506 of FIG. 5;

FIG. 8 is a flow diagram of a method of extracting a watermark from MPEG media data;

FIG. 9 is a flow diagram of a method of extracting a watermark from uncompressed media;

FIG. 10 is a flow diagram of a method of detecting a watermark from MPEG data, with registration; and

FIG. 11 is a flow diagram of a method of detecting a watermark from uncompressed image data, with registration.

DETAILED DESCRIPTION

As used in the following description the terms image and image data will be understood to be equally applicable to video, image and multimedia data. The term "watermark" will be understood to include embedded data, symbols, images, instructions or any other identifying information.

In order to better understand the present invention, first a review of the basic watermarking method will be presented followed by additional descriptions of the improvements comprising the present invention.

First, we define some notations. Let a watermark to be embedded into an image be an N dimensional vector, denoted by W[1, . . . ,N]. In the following text, the notation W[1, . . . , N] is used in the same manner as W[k] (k=1, . . . , N). Let V[1, . . . ,N] denote a vector value extracted from an image, where the element V[k] corresponds to W[k]. Specifically, the value V[k] is a weighted sum of DCT coefficients given by

$$V[k]=D\_k[1]F\_k[1]+D\_k[2]F\_k[2]+\ldots+D\_k[n\_k]F\_k[n\_k]$$

where D_k[i](i=1, . . . , n_k) indicate members of the set of DCT coefficients used for calculating V[k], n_k indicates the number of members, and F_k[i](i=1, . . . , n_k) are weighting coefficients related to a filter processing. The concept of F_k is that the DCT coefficients are weighted according to how much noise might be expected in each coefficient. To calculate V[1, . . . ,N], nxn DCT coefficients are first calculated over a whole image. Then the coefficients are classified into N sets, each of which is related to each element of V[1, . . . , N]. The rule of classifying DCT coefficients is predetermined, and the same rule is used in both inserting and detecting a watermark.

Before inserting a watermark into an image, a detection algorithm is applied to the image to find a watermark that is already present in the image. If the image does not contain a watermark, the extracted values, V[1, . . . , N] will be normally distributed random numbers that do not correlate any watemrark W[1, . . . , N]. A watermark W[1, . . . , N] is inserted into an image by changing each of D_k[1, . . . , n_k] (k-1, . . . , N) slightly in order to make the extracted value V[1, . . . , N] highly correlate the watermark W[1, . . . , N]. Let the target value of V[1, . . . , N] be denoted by ω[1, . . . , N], that is, the values V[1, . . . ,N] are changed to ω[1, . . . , N] by inserting the watermark W[1, . . . , N]. The target values ω[1, . . . , N] have high correlation with the watermark W[1, . . . , N] and they are determined as will be described below.

After the target values ω[1, . . . , N] are determined, the difference ω[k]−V[k] is distributed among the DCT coefficients D_k[1, . . . , n_k]. Then a watermark is inserted by adding the allocated difference value to the corresponding DCT coefficients D_k[1, . . . , n−k]. This change of DCT coefficients must be done in such a manner so as not to change the appearance of the image.

In distributing the difference ω[k]−V[k], a characteristic of the human visual process is taken into account. The amount of change that does not cause a visible change in the image is different for each DCT coefficient D_k[i]. This amount depends on the human visual process which can be approximately simulated with a computational model. The amount of change is referred to as "slack". The slack is calculated for each DCT coefficient and it is used in distributing the difference ω[k]−V[k] among the DCT coefficients. Next we describe how to calculate the values of the slack using a model of the human visual process.

The preferred computational model of human visual sensitivity that is used in the present invention is found in an article by Andrew B. Watson, entitled "DCT Quantization Matrices Usually Optimized for Indirect Images" in SPIE, vol. 1913 (1993), pp. 202–216. This model was applied to watermarking in an article by Christine I. Podilchuk and Wenjun Zeng entitled "Digital Image Watermarking Using Visual Models", Proc. of EI'97, vol. 3016, Feb. 9–14, 1997. The current invention differs from that of Podilchuk and Zeng in (i) not requiring the original unwatermarked image at the decoder and (ii) not extracting the watermark from the individual 8×8 blocks, but from the sum of a set of 8×8 blocks. Other computational models are also usable.

For each element of the image's block DCT, d[i,j], this model computes a value called the element's "slack", S[i,j], which indicates how much a particular d[i,j] value may be altered before such an alteration becomes visible. The value is computed in three steps. The first step models the contrast masking phenomenon of the human visual system and models the visual sensitivity at different frequencies and handles the difference between visual sensitivity to changes in different frequencies. The second step models the luminance masking phenomenon of the human visual system and handles the fact that the visual system is more sensitive to changes in dark regions than to changes in bright regions. The third step handles the fact that the sensitivity to changes depends in part on the percentage that the frequency is changing (i.e. a DCT term with a small value in it may only change a little, while one with a larger value may change more).

The perceptual model makes use of a matrix of values that indicate the relative sensitivity of the human visual system to the different terms of a spatial 8×8 DCT. The formulae for computing this matrix are available in an article by Albert J. Ahumada Jr. and Heidi A. Peterson entitled "Luminance-Model-Based DCT Quantization for Color Image Compression", in SPIE, vol. 1666, (1992) pp. 365–374.

After computing the slacks for all the 8×8 DCT's in the image, a slack can be assigned to each D_k[1, . . . ,n_k]. Call these slacks S_k[1, . . . ,n_k]. It is now possible to distribute the changes in the V's over all the D_k's with minimal visual impact. This is done according to the following formula:

$$D'\_k[i] = D\_k[i] + \frac{(\omega[k] - V[k]) * S\_k[i]}{\sum_{j=1}^{n\_k} S\_k[j]F\_k[j]}$$

where D'_k[1, . . . , n_k] are the modified 8x8 DCT coefficients, and ω[k] and V[k] are the k'th elements of ω and V, respectively. The effect of this formula is to distribute the desired change in a given element of the watermark vector (ω[k]−V[k]) over all the DCT coefficients that are summed to produce that element, proportionately according to those DCT coefficients' slacks. To illustrate, consider two simple examples: 1) If all the slacks are 0 except for slack S_k[m], then the sum of all the values S_k[j]F_k[j] is equal to S_k[m]F_k[m], and only D_k[m] is changed. It is changed by the full value of (ω[k]−V[k]). 2) If all the slacks are equal and all the coefficients F_k[i]=1, then each D_k[i] is changed by the same amount.

After making these changes, convert all the 8x8 DCT's back into the spatial domain, and the result is a watermarked image. It is easy to show that the sum of all the D_k's for a given k will equal ω[k]. The process of making this is referred to as "inserting Omega into the image". The watermark extracted from the resulting image, if the image has not been attacked, will be exactly ω, not ω plus noise.

There are two important issues remaining to be discussed. First, how to decide on ω, and, second, how to make the watermark robust.

Previously, the equivalent of ω was computed as:

$$\omega = V + \alpha * W$$

where α is a small constant, and W is a zero-mean watermark signal. It is possible to use the same formula here, but it is too limiting to result in the strongest possible watermark using the present invention. In practicing the invention, it is often possible to insert an ω that has perfect correlation with the watermark, W, without causing any visible change in the image. The following formula is used:

$$\omega = mean(V) + \beta * (V - mean(V)) + \alpha * W$$

This result is a weighted sum of the watermark signal and the original, noise (image) signal. If β is set to 0, the result is an ω that perfectly correlates with W.

The signal to noise ratio for an unattacked image will be:

$$SNR = \alpha * std(W) / \beta * std(V)$$

where std(X) is the standard deviation of X.

There are many ways to choose α and β based on optimizations to maximize different criteria such as fidelity or robustness.

At this point there is a complete method of inserting watermarks. The method contains explicit modeling of human vision, but it does not contain any explicit method of making the watermark robust. In fact, the method as described so far will try to put as much of the watermark as possible into the high frequencies since these frequencies have the largest slack, but this is a poor thing to do from the point of view of robustness.

To make the watermark robust against a given set of attacks or signal degradations, it is first necessary to consider how those attacks affect the various terms of the 8x8 DCT's in the image. Then, terms that are affected by attacks or signal degradations in similar ways are grouped together, and watermarked as if the group of terms were a separate image.

The following is a simple example. Suppose there is only concern about two possible attacks: cropping 24 columns of pixels from the left side of the image, or cropping 24 columns of pixels from the right side of the image. This results in three groups of DCT terms: those that come from the 3 left-most columns of 8x8 DCT's blocks, those that come from the 3 right-most columns, and those that come from the rest of the image. All the terms in each of these groups either survives or is destroyed by any given attack together. If each group is watermarked as though it were a separate image, then the watermark from at least one group will generally survive attack (assuming that the 24-column cropping attacks are the only attacks possible), and the watermark that is extracted will consist of the correct watermark, from that group, plus some noisy watermarks from groups that were damaged by the attack.

A more interesting example is low-pass and high-pass filtering attacks. It is possible to group all the low frequencies together into one group, and all the high frequencies into one or more other groups. If the predetermined rule for classifying DCT coefficients into the N sets is designed in such a way that each set has coefficients of many different frequencies, then the complete watermark can be inserted into each group. Then, if the high frequencies are removed, the watermark will still be detectable in the low frequencies, and vice-versa.

The more groups the terms are divided into, the more robust will be the watermark. There is a cost because it will become increasingly more difficult to distribute the changes without causing visible distortion.

It is important to note that the best balance can be achieved after the detectors are in wide use. It is possible to modify the insertion algorithm to make watermarks robust against a wide variety of attacks without having to change detection at all.

Presently only three groups are used. One group collects together most of the low frequencies. Each of the other two groups represents one higher frequency.

In the preferred method, a watermark is not placed in any of the higher frequencies. The reason is not that they are susceptible to attack (that is handled by the design of the Filter [ ] used in detection). Rather, it is because watermarking the higher frequencies causes MPEG compression rates to go down substantially. An alternative solution might be to add other groups that contain higher frequencies.

In FIG. 1, there is shown a flow diagram of the watermark insertion method. The digital image is divided into a collection of nxn blocks, preferably 8x8 blocks, in step 10. The discrete cosine transform (DCT) of each block is computed in a known manner in step 12. The DCT's are separated into groups that respond to different attacks in the same manner 14. A first group G is selected in step 16.

Next, extract a watermark V, using only the terms in the group G in step 18. Determine a new signal (target value) ω selected such that ω is similar to V but is highly correlated with watermark W in step 20.

Add fractions of ω−V to terms in G according to perceptual slack in step 22. Decide whether group G is the last group in step 24. If not, select next group G in step 26, and extract watermark V using only terms of next group G in step 18 and continue procedures until the last group G is found in step 24. Next, compute the inverse DCTs of the blocks in step 28 resulting in a watermarked image.

Some alternative steps in the insertion method are possible. For example, the distribution of the difference between ω and V over DCT terms can be done stochastically to help deter tampering and reduce susceptibility to tampering.

Also, the groups of DCT terms for robustness purposes could be performed dynamically. For example, the insertion program could simulate various attacks on the image and determine the effect on the values of the DCT term. Then, the program would cause appropriate allocation of the terms into the groups similarly affected.

The inserter can be designed with a user-interface that allows the user to set two parameters: (1) the maximum perceptual difference between the original image and the watermarked image (J) and (2) the maximum allowable probability of missed detection after any of the predefined set of attacks. The algorithm would then insert watermarks into a large number of images automatically, according to the allowable perceptual change J and checking each one against simulations of the attacks. If an image fails to meet the specified robustness constraint (maximum allowable probability of missed detection), then the user would be notified so that a manual decision can be made to compensate or trade-off image fidelity for robustness.

In addition, the distribution of the difference between ω and V over DCT terms can be modified to explicitly compensate for MPEG quantization. Using the above watermark insertion method there may result a degraded watermark in the watermarked data. In order to enhance the watermark in the watermarked data after MPEG compression several techniques are possible.

FIG. 2 is a schematic diagram of a typical MPEG-2 encoder. FIG. 2 depicts elements which are indispensable to execute an MPEG-2 encoding of P pictures, or to perform a combined interframe prediction and DCT coding. Input images are provided as one input to subtractor **30**. The other input to subtractor **30** is predicted image generated in frame memory **32**. The predicted images are subtracted from the input images at subtractor **30**. A discrete cosine transform (DCT) is performed at DCT calculator **34** on the output signal from subtractor **30**. The DCT coefficients are quantized in quantizer **36**. The outputs of the quantizer **36** are sent to a variable length encoder **38** where Huffman encoding is performed. The quantized DCT coefficients outputted from the quantizer **36** are also sent to an inverse quantizer **40** where they are de-quantized. Inverse DCT of the de-quantized DCT coefficients is performed in the inverse DCT calculator **42**. The results are added at adder **44** to the predicted image outputted from the frame memory **32**, and then an image which is expected to be the same as that acquired in a decoder is reconstructed. The reconstructed image is called "a locally decoded image." This locally decoded image is stored in the frame memory **32** to produce the predicted images.

FIG. 3 is a schematic diagram of a modified MPEG-2 encoder for reducing degradation of a watermark in watermarked data. Before MPEG-2 encoding, DCT is performed on a input image at the DCT calculator **50** and watermark signals are added to the DCT coefficients at adder **52**. The output DCT coefficients including watermark information is subject to inverse DCT in the inverse DCT calculator **54**. The output of inverse DCT calculator **54** are images with a watermark. These watermarked images are sent to MPEG-2 encoder and MPEG-2 encoding is performed as described above. In addition, in this embodiment, watermark information is modified in order to be suited to MPEG-2 compression. DCT coefficients for the predicted images are calculated in DCT calculator **56**. The quanitization values outputted from quantizer **36** are de-quantized in inverse quantizer **58**. The results of inverse quanitization are added at adder **60** to the DCT coefficients outputted from DCT calculator **56**. The results of addition correspond to the DCT coefficients for the decoded images which are expected to be generated in a decoder. These DCT coefficients are inputted into a watermark correction device **62**. The watermark correction device **62** outputs watermark correction signals. At adder **64**, the watermark correction signals from device **62** are added to the quantization values from quantizer **36**. The output of adder **64** is used as the inputs to variable length encoder **38** and inverse quantizer **40**.

Next, we describe the process performed in the watermark correction device **62**. Let us introduce several new notations to explain the process. Let Dq_k[i] be the quantization value corresponding to D_k[i], that is, the quantization value of the i-th member of the k-th set for calculating the value V. Let Q_k[i] be the quantization step size used in obtaining Dq_k[i]. Let Dr_k[i] be the output value of adder **60** that is obtained by adding the inverse quantization value of Dq_k[i] calculated in inverse quantizer **58** to the corresponding DCT coefficient outputted from DCT calculator **56**. Let Vr[1, . . . ,N] be the value extracted from the output values of adder **60**, Dr_k[i], in the same manner that the value V[1, . . . , N] is calculated in inserting a watermark. We assume that the target value ω[1, . . . , N] is also available in the watermark correction device **62**. FIG. 5 shows that the flow chart describing the process performed in the watermark correction device **62**. First the index k of the watermark element is set to 1(Step 500). Next the value Vr[k] is calculated (Step 502) by

$$Vr[k]=F\_k[1]Dr\_k[1]+ . . . F\_k[n\_k]Dr\_k[n\_k],$$

where the weighting coefficients F_k[1, . . . ,n_k] are the same value as those used in calculating V[k]. Then the absolute value of the difference between the value Vr[k] and the target value ω[k], and the sign of the difference are computed in step 504 by the following equations:

$$Dif=|Vr[k]-\omega[k]|$$

$$s=SIGN(Vr[k]-\omega[k]),$$

where

$$\mathrm{Sign}(x)=\begin{cases}1 & (x\geq 0)\\ -1 & (x<0)\end{cases}$$

This value Dif corresponds to the distortion of the watermark inserted at the adder **52** generated in the quantization process.

On the basis of the absolute value Dif and the sign s watermark correction signals are generated in step 506. The process in Step 506 is described below. After calculating the watermark correction signals for the DCT coefficients related to the k-th element of the watermark, index k is compared with N in step 508. If k>N. then the process is finished. If k≦N, the value k is increased by one in step 510, and the process goes back to step 502. The watermark correction process is thus performed, and the obtained watermark correction signals are finally outputted to adder **64**.

Next, we describe the process performed in step 506 in FIG. 5, using the flow chart in FIG. 6. In step 506, the watermark correction signals for the DCT related to the k-th element of the watermark are generated. The array ΔDq_k [i](i=1, . . . n_k) are first all set to zero in step 520. Next, the value j is set to one in step 522. Then the index of the DCT coefficients i is found through a permuting function p(j) in step 523. The function p(j) returns the j-th value of a permutation obtained after the integers 1 to n_k are per-

muted. The simplest example is p(j)=j. Next, a value −s is stored in ΔDq_k[i], and the value Dif is decreased by Q_k[i]F_k[i] in step 524. This indicates that the quantization value Dq_k[i] is changed by −s by adding ΔDq_k[i] to Dq_k[i] at adder **64**. The value s is 1 or −1, so the change in the quantization value is one. In other words, de-quantized value obtained in a inverse quantizer in a decoder is changed by −Q_k[i], one step size. The value Dif after the update is identical to the absolute value of the difference between ω[k] and Vr[k] calculated with the corrected quantization values Dq_k[i]+ΔDq_k[i](i=1, . . . n_k). After step 524, the values Dif and zero, and index j and n_k are compared in step 526. If Dif<0 or j>n_k, then this subroutine is finished. If the condition is not satisfied, the index j is increased by certain amount Δj in step 528, then the process returns to step 523.

Instead of the process shown in FIG. **6**, an alternative process shown in FIG. **7** can be used as the process of step 506. In the process shown in FIG. **7**, a step 530, checking whether the quantization value Dq_k[i] equals zero or not, is added between step 523 and step 524. In this case, step 524 is performed only if the quantization value Dq_k[i] is a non-zero value. This allows a reduction in the increase of the number of bits caused by correcting watermark information, because changing a quantization value from zero to non-zero value generally results in a large increase in the number of bits.

FIG. **4** is a schematic diagram of an alternative embodiment of a modified MPEG-2 encoder for reducing degradation of a watermark in watermarked data. In this embodiment, the basic concept is the same as the described in connection with FIG. **3**. The differences lie in the fact that subtraction of the predicted images from the input original images is performed in the DCT domain not in the spatial domain. For the predicted image outputted from the frame memory **32**, DCT is performed in DCT calculator **70**, and the results are subtracted from the watermarked DCT coefficients at subtractor **72**. The results of subtraction are sent to the quantizer **36** and then the watermark correction is performed in the same manner as shown in FIG. **3**. The results outputted from subtractor **72** are the same as the results outputted from the DCT calculator **34** in FIG. **3** because of the linearity of DCT. Therefore, the results obtained in the processes followed by quantization in quantizer **36** are the same as those in FIG. **3**. This embodiment results in a reduction in the number of DCT calculations.

The detection procedure to detect a watermark in an image will now be described.

If MPEG video is the input image data format, the following detection process determines whether watermark W is present, where W[1, . . . , N]=the watermark being tested for.

Decode the Huffman code, but do not compute the inverse DCT's, so that, for each frame (at least, each I-frame), there is an array of 8×8 DCTs.

Next perform the same summation of DCT coefficients that was performed during watermark insertion to obtain the vector V. Compute the correlation coefficient C, between V and the watermark being tested for, W:

$$V' = V - \bar{V}$$

$$W' = W - \bar{W}$$

-continued

$$C = \frac{W' \cdot V'}{\sqrt{(W' \cdot W')(V' \cdot V')}}$$

Finally, convert C into a normalized Fisher Z statistic:

$$Z = \frac{\sqrt{N-3}}{2} \log \frac{1+C}{1-C}$$

where N is the length of the watermark.

The Z value indicates whether the watermark is present. A preferred threshold for Z is 4 (i.e. Z≧4 means the watermark is present), but other values may be used depending on the desired probabilities of false alarms and missed detections.

FIG. **8** is a flow diagram of the detection method for MPEG video input described above. The input MPEG video is subject to a Huffman decoder and partial parser **80** where the output is a set of DCT for n×n, preferably 8×8, blocks of the video input.

The n×n DCTs are provided to watermark accumulator **82**. Accumulator **82** has memory whose length is the watermark length N. DCT coefficients from the Huffman decoder and partial parser **80** are classified according to a predetermined rule and summed for extracting a watermark as mentioned before, and the results are accumulated in the memory. The extracted watermark is proved to comparator **84** where it is compared with possible watermarks in the image by calculating correlation coefficients between the extracted watermark and the possible watermarks as mentioned before. The possible watermarks are the universe of the watermarks accumulators and comparator are found, for instance, in U.S. patent application Ser. No. 08/746,022.

The output of comparator **84** is the likelihood (normalized Fisher Z statistic) of the detected watermark being each of the possible watermarks. The most likely watermark is determined and is deemed the watermark in the image, or, if the detector does not exceed a predetermined threshold, then no watermark is present.

Alternatively, if the incoming input data comprises an uncompressed image, an embedded watermark can be detected by applying the method above to DCT coefficients obtained by performing 8×8 DCT for the whole image. In this case, DCT have to be performed for each 8×8 block, but a skillfully designed rule for classifying DCT coefficients into N sets enables us to avoid performing DCT many times. Before explaining the method to reduce DCT calculation, let us define some notations.

Let h_m(i, j) (i=1 . . . 8,j=1, . . . 8. m=0, . . . ,M−1) be a set of functions that map frequency indices of 8×8 DCT coefficients (i, j) onto the indices k of the element of a watermark, and M indicates the number of functions. So if k=h_m(i, j), then a DCT coefficient whose index is (i, j) is classified into the k-th set of DCT coefficients for calculating the value V[k]. We prepare M different functions h_m(i,j) (m=0, . . . M−1). Which function is selected for a certain 8×8 block depends on the numbers r and c of the block where r and c indicate the row and column numbers of the block respectively. So the index of the functions, m, is first found according to the values r and c, then the index of the sets, k, is determined by h_m(i,j) for each DCT coefficient. Using these functions h_m(i,j) in classifying DCT coefficients is assumed in the remaining part of the detailed description.

In this case, we can reduce the number of DCT calculations in the following manner. First the sum of the blocks whose indices m are the same is computed for each

m=0, . . . , M−1. Let this summed block be denoted by VB_m[i,j](m=0, . . . ,M−1). Then DCT is perfomred for the M summed blocks VB_m[i,j](m=0,. . . , M−1). Finally, the DCT coefficients of the summed blocks are classified into N sets according to the value h_m(i,j), and added together within each set to obtain V[1, . . . ,N]. The obtained results V[1, . . . N] are the same as V[1, . . . N] obtained with the method mentioned above because DCT is a linear transform, that is, the sum of DCT blocks equals the result of DCT for the sum of the blocks. If M is much less than the total number of blocks in an image, the number of DCT calculations is dramatically reduced. This method thus allows us to extract watermarks with small calculation cost.

FIG. **9** shows a flow diagram of the detection method for uncompressed video input data as described above.

The uncompressed video image data is provided to n×n accumulators, preferably 8×8 accumulators **90**. The memory requirement is n^2 times the number of the function h_m(i,j). For each index m. the blocks with the same index m are summed and the resultant M summed blocks are accumulated in the memory.

The output is the summed signal of each of the n×n blocks. The output is subject to a DCT transform **92**. The number of transformations is proportional to the number of the functions h_m(i,j). The result is a group of n×n DCTs which is classified into N sets according to the functions h_m(i,j) and summed for extracting a watermark as mentioned before. The obtained watermark is provided to watermark accumulator **94** and accumulated. The memory requirement for accumulator **94** is proportional to the watermark length. The extracted watermark is provided as input to comparator **96**. The other inputs to comparator **96** are the possible watermarks that may have been inserted into the input image data. The comparator computes a likelihood (normalized Fisher Z statistic) of each possible watermark having been inserted into the image data. The most likely watermark is determined and is deemed the watermark in the image.

A limitation of block based DCT methods is their sensitivity to spatial shifts of the image. For example, if the image is shifted two pixels to the right, then the DCT coefficients change significantly, so that the watermark cannot be detected. Furthermore, general distortions, such as scaling and rotation, also make the watermark undetectable.

To solve these problems, the above insertion and extraction methods may be modified in two ways. The first possible modification is to insert multiple watermarks designed to survive predefined distortions of the video. The second modification is to arrange that translations can be compensated for without performing the summation more than once. Optionally, this second modification may be further modified to insert registration patterns, one for each of the multiple watermarks, which can be used by a modified watermark detector to compensate for arbitrary translations of the video.

When detecting watermarks inserted using the above method, it is necessary to divide the image into the same grid of n×n blocks as was used during insertion. If the image has been translated since watermark insertion, then determining the correct grid becomes difficult. In many applications, this is a serious problem, since certain, specific transformations can be expected. For example, it can be expected that video on a DVD disk might be modified to fit on a standard television screen by conversion to either "panscan" or "letterbox" mode. In "panscan" mode, the horizontal image resolution is increased, and the image is cropped at a predetermined offset, so that the resulting image will be correct when viewed on a 3×4-aspect-ratio television screen. In "letterbox" mode, the image is scaled vertically, and black is added at the top and bottom, so that the whole image will fit correctly on a 3×4-aspect-ratio screen. Since these two geometric transformations are more likely than any other, it is reasonable to prepare for them specifically.

The problem of the predetermined scaling or transforming of watermarked video is solved in the present invention by inserting an additional watermark for each of the likely transformations. Each of these watermarks is designed so that, when the image has undergone the corresponding known transformation, the grid of n×n blocks used during insertion will align with a predetermined grid used during detection. Thus, if the image has undergone no transformation, then the detection grid will align with the normal mode watermark, and the normal mode watermark will be detected. If the image has undergone "panscan" transformation, then the same detection grid will align with the "panscan" watermark, and the "panscan" watermark will be detected, and so forth for "letterbox" scan or any other predefined transformation.

The procedure for inserting a watermark that is to be detected after a specific transformation comprises the following steps:

1. Make a copy, I_T, of the image being watermarked, I, and apply the transformation to be compensated for the image. For example, I_T might be a copy of I that has been transformed into "letterbox" mode by vertical shrinking of I.
2. Create a watermarked version of the transformed image, I_T', according to the general watermarking method described above.
3. Let $W_T=I_T'-I_T$ be the spatial pattern that was added to I_T when it was watermarked.
4. Perform the inverse transformation on W_T to yield the corresponding watermark pattern, W, for the untransformed image. For example, if the transformation to be compensated for was "letterbox" mode, then W would be obtained by vertically expanding W_T.
5. Let I'=I+W be the image with a watermark added for the given transformation.

When the transformation is applied to I', the result will be approximately I_T', and the watermark will be detected by the same procedure designed to detect a normal watermark.

This process can only be used for a small number of transformations, as each additional watermark causes additional degradation of the image, and reduces the detectability of other watermarks in the image. However, tests have shown that three watermarks—two for transformed images and one for the untransformed image—result in acceptable fidelity and good detectability. Alternatively, for video, each watermark can be inserted in a time-multiplexed manner.

In cases where the transformations that an image will undergo are not predefined or predetermined, or where there are too many probable transformations to allow for the insertion of a separate watermark for each transformation, the above described method of compensating for transformations is not optimal. Thus, the present invention includes an additional improvement, which compensates for arbitrary translation of the image between the times of watermark insertion and watermark detection.

Arbitrary translation is compensated for in two ways. One way is by translations by even multiples of 8 pixels in the x or y directions when 8×8 blocks are used. This can be easily compensated for if the following restrictions are imposed on the relationship between an index of h_m(i,j), m, and the

row and column numbers of blocks, r and c. We determine the index m according to

$$m=f(r,c) \bmod M,$$

where f(r,c) is a linear function of r and c, and f(0,1) and f(1,0) are integers. In addition, let h_m(i,j) be a function expressed as

$$h\_m(i,j)=(h\_0(i,j)+a\times m) \bmod N,$$

where a is an integer. These restrictions are assumed in the remaining part of the detailed description. In this case, the shift compensation is performed in the basic detection algorithm by computing the correlation of the extracted watermark with all cyclical shifts of the watermark being tested for. This is because the values VB_m[i,j] (m=0, . . . , M−1) obtained from a watermarked image shifted by a multiple of 8 pixels in the horizontal and/or the vertical directions are identical to the values VB_m[i,j](m=0, . . . , M−1) obtained from the non-shifted image except that the indices m are cyclically shifted. As a result, the extracted value V[1,. . . ,N] obtained from a watermarked image shifted by a multiple of 8 are identical to the value V[1, . . . , N] obtained from the non-shifted image except that the elements V[k] are cyclically shifted. For n×n grid format, the same thing is true of a shift of a multiple of n in the horizontal and/or the vertical directions.

For shifts of less than 8 pixels in x and/or y directions, an exhaustive search can be performed of all 64 possibilities and the maximum Z value taken from the set of 64×M−Z values, where M is the number of 8×8 accumulators. In our tests M was chosen to be 64. The factor M is necessary to account for the cyclic shifts that are introduced by shifts of a multiple of 8 pixels.

The exhaustive search requires shifting the M 8×8 accumulator array in the spatial domain and then performing the DCT of each of the M 8×8 blocks. This is performed 64 times, once for each of the possible shifts. Thus it is necessary to perform 64×M 8×8 DCTs. If this computation is too expensive in terms of time or memory an alternative method can be used, as described below.

The second way, which compensates for translations of non-even multiples of n pixels, uses a pattern (referred to as a "registration pattern") which can be inserted at the time of watermark insertion. By finding the location where the registration signal best matches a predefined signal, a detector can determine how much to shift the data before extracting the watermark. This shifting must be done in the spatial domain, but can be done with accumulators, so conversions of whole images are avoided.

Moreover, the 64×M 8×8 DCTs are unnecessary. Instead, the correct registration is determined in the spatial domain and then compensated for by shifting the pixels in the accumulator arrays. The M 8×8 accumulators are only then transformed into the DCT domain and the watermark extraction is performed as described above.

The registration pattern is an 8×8 spatial pattern inserted into the image in such a way that the sum of all 8×8 pixel blocks highly correlates with the pattern. Again, an n×n spatial pattern is used when the video is n×n blocks. A registration pattern can be inserted by using the watermark insertion method described above. The sum of all 8×8 pixel blocks becomes highly correlated with a registration pattern if the DCT coefficients of the sum block and those of the registration pattern are highly correlated to each other. In addition, the DCT coefficients of the sum block equals the sum of all DCT blocks because of the linearity of DCT. We can thus insert a registration pattern using a method similar to that inserting a watermark described above, considering the DCT coefficients of a registration pattern as a watermark W and considering the sum of all 8×8 DCT blocks as a value V.

Insertion is performed by converting the registration pattern into the DCT domain, and then using the basic insertion algorithm with different D_ks and F_ks. Specifically, each AC term of the registration pattern's DCT is considered one element of the watermark, W[k]. The set of DCT terms that are summed together to extract this element, D_k, is simply the set of corresponding terms of all the 8×8 DCTs in the image. All the F_ks are set to 1. Using these D_ks and F_ks, the insertion algorithm inserts a registration pattern along with each watermark. The watermarks are still inserted with the original D_ks and F_ks.

During detection, registration is performed as follows. Each of a predetermined number of blocks is summed together to form a single n×n, typically 8×8, block. We have arbitrarily used 64 blocks in our tests. This block contains a registration pattern placed there by the insertion process. To determine the horizontal and vertical translation of the frame, a correlation process is performed in the spatial domain to determine these offsets. Sixty-four correlations are performed for each of the 8 horizontal and 8 vertical motions that are possible. When the 8×8 patch is shifted either horizontally or vertically, a wrap around shift is performed.

We now describe how to determine the shift of the grid. In the following method, we assume the integer values f(0,1) and f(1,0) are relatively prime to M.

First, the blocks for which the same function h_m(i,j) is used in classifying DCT coefficients are added together in the spatial domain for generating a summed block VB_m[i,j] for each m. The sum of all n×n blocks denoted by AB[i,j], is computed by

$$AB[i,j]=\sum_{m=0}^{M-1} VB\_m[i,j] \quad (i=1, \ldots, n,\ j=1, \ldots, n).$$

Then the correlation coefficient between AB[i,j] and a registration pattern, denoted by R[i,j] is computed. After calculation of the correlation coefficient, the values AB[i,j] are cyclically shifted by one column in the horizontal direction, and the correlation coefficient between AB[i,j] and R[i,j] is calculated in the same way. The same operations are repeated for each shift. After shifting n times, AB[i,j] becomes identical to AB[i,j] before any shift is done. Then, the values AB[i,j] are cyclically shifted by one row in the vertical direction, and calculation of a correlation coefficient and shift by one column in the horizontal direction are repeated. In this way, we can calculate correlation coefficients for all n^2 possible shifts. At the same time, we search the shift value (offset), denoted by (X,Y), which gives the maximum correlation coefficient.

After the offset (X,Y) has been determined, the M summed blocks VB_m[i,j] are then shifted accordingly in the spatial domain. Next, we describe the method to compensate for the shift value X in the horizontal direction. To do so, the values VB_m[i,j](m=0, . . . ,M−1) are first copied on an n×nM array VB1[i,j] (i=1, . . . ,n.j=1, . . . ,nM). copying the values VB_m[i,j], the spatial relationships between blocks, that is, which blocks are adjoining a certain block are considered. The function f(r,c) is linear, so

$$f(r,c+1)=f(r,c)+f(0,1).$$

This indicates that the blocks whose index m equals

$$m1=(m0+f(0,1))mod\ M$$

are located next to the blocks in which m=m0. So, the values VB_m1 are copied next to the values VB_m0 in the array VB1. For this reason, for each I (I=0, . . . ,M−1), the values VB_m'[i, j](i=1, . . . ,n,j=1, . . . , n, m'=I×F(0,1)mod M) are copied in the n×n square region of VB1 where the index of the left-top corner is (1, nI+1). After copying the values for all I, the array VB1 is filled with the values VB_m[i,j] because f(0,1) and M are relatively prime. Next, the values in the array VB1 are cyclically shifted by X in the horizontal direction. Then the values in VB1 are returned to VB_m in such a way that the value in the n×n region of VB1 where the index of the left-top corner is (1, nI+1) are substituted to VB_m'[i,j](i=1, . . . ,n, j=1,. . . ,n, m'=I×f(0,1) mod M) for each I(I=0, . . . ,M−1). The horizontal offset value X can thus be compensated for without shifting the whole image itself.

Next, the shift value Y in the vertical direction is compensated for. To do so, the values VB_m[i,j](m=0, . . . , M−1) are first copied on an nM×n array VB2 [i,j] (i=1, . . . , nM, j=1, . . . ,n). In copying the values VB_m[i,j], the spatial relationship between blocks are considered. The function f(r,c) is linear, so

$$f(r+1,c)=f(r,c)+f(1,0).$$

This indicates that the blocks whose index m equals

$$m1=(m0+f(1,0))mod\ M$$

locate under the blocks in which m=m0. So, the values VB_m1 are copied under the values VB_m0 in the array VB2. For this reason, for each I(I=0, . . . ,M−1), the values VB_m'[i,j](i=1, . . . ,n, j=1, . . . , n, m'=I×f(0,1) mod M) are copied on the n×n square region of VBI where the index of the left-top corner is (nI+1, 1). After copying the values for all I, the array VB2 is filled with the values VB_m[i,j] because f(1,0) and M are relatively prime. Next, the values in the array VB2 are cyclically shifted by Y in the vertical direction. Then the values in VB2 are returned to VB_m in such a way that the value in the n×n region of VB2 where the index of the left-top corner is (nI+1, 1) are substituted in VB_m'[i,j](i=1, . . . n, j=1, . . . ,n, m'=I×F(0,1) mod M) for each I (I=0,. . . ,M−1). The vertical offset value Y can thus be compensated without shifting the whole image itself.

The offset of the n x n grid is compensated by the process mentioned above. These processes are performed in the registration process **108** as will be explained later. A shift of a multiple of n remains even after these processes, but this shift does not affect the watermark detection because the correlation coefficient between a watermark W and an extracted value V is calculated by shifting the watermark W cyclically as described above. After the registration process above is applied, the M blocks VB_m(m=0, . . . ,M−1) are transformed back to the DCT domain.

With reference now to FIGS. **10** and **11**, there are shown the basic detection algorithms modified to compensate for translational registration. In the case of MPEG video input (FIG. **10**), 8×8 DCT blocks obtained from an MPEG video stream are first classified into M groups according to their indices m of the function h_m(i,j), summed within the groups for generating M summed blocks, and the resultant summed blocks are accumulated in 8×8 accumulators **102**. The M summed blocks in accumulators **102** must be converted into the spatial domain by performing an inverse DCT operation in inverse DCT converter **104**, and accumulated in accumulators **106**. Finding the offset value of the 8×8 grid and compensating for the offset is executed for the output from 8×8 accumulators **106** in registration **108** as described above. The registration data outputted from registration process **108** is accumulated in accumulators **110** and converted into the DCT domain in DCT converter **112** for watermark extraction by use of accumulators **114**, watermark extractor **116** and watermark decoder **118**. In watermark extractor **116**, the DCT coefficients outputted from accumulator **114** are classified into N sets according to the functions h_m (i,j) and summed for extracting a watermark. The obtained watermark is provided to watermark decoder **118**, in which the processes executed in comparator **84** in FIG. **8** for finding a watermark corresponding to the extracted watermark. The watermark considered to have been inserted is outputted from the watermark decoder **118**. In the case of uncompressed input data (FIG. **11**), the input data is divided into 8×8 blocks and accumulated in accumulators **106** according to the indices of the functions h_m(i,j), and registration **108** is performed before conversion into the DCT domain in DCT converter **112**. The process continues as described above.

While there has been described and illustrated methods of insertion and detection of watermarks in image data, it will be understood by those skilled in the art that variations and modifications are possible without deviating from the spirit and broad teachings of the present invention which shall be limited solely by the scope of the claims appended hereto.

What is claimed is:

1. A method of inserting a watermark signal into video images comprising the steps of:

receiving input video images;

performing discrete cosine transformation (DCT) of said input video images to obtain DCT values of said input video images;

adding watermark signals to said DCT values to obtain DCT values with watermark;

performing an inverse DCT on the DCT values with watermark for generating watermarked images;

subtracting predicted images from said watermarked images for generating residual images;

obtaining the DCT values of residual images and quantizing the DCT values of the residual images;

inverse quantizing the quantized DCT values of the residual images;

performing discrete cosine transformation of the predicted images;

summing the inverse quantized DCT values of the residual images and the DCT of the predicted images for generating DCT coefficients for decoded images;

calculating correction signals from the DCT coefficients of the decoded images and adding the correction signals to the quantized DCT values of the residual images for obtaining an output signal;

inverse quantizing the output signal, inverse DCT the inverse quantized output signal and summing the resultant signal with the predicted images for generating a summed signal;

storing the summed signal in memory for generating the predicted images; and

variable length encoding the output signal for providing a watermarked MPEG video signal of the input video images.

2. A method of inserting a watermark signal into video images as set forth in claim **1**, where said calculating step generates a negative value as said correction signal when the

extracted value exceeds the corresponding target value, and generates a positive value as said correction signal when the extracted value falls short of the corresponding target value.

3. A method of inserting a watermark signal into video images as set forth in claim 2, where said positive value is +1 and said negative value is −1.

4. A method of inserting a watermark signal into video images as set forth in claim 2, where said calculating step generates a zero value as the correction signal if the corresponding quantized DCT values of residual images are zero.

5. A method of inserting a watermark signal into video images as set forth in claim 4, where said positive value is +1 and said negative value is −1.

6. A method of inserting a watermark signal into video images comprising the steps of:

receiving input video images;

performing discrete cosine transformation (DCT) of said input video images to obtain DCT values of said input video images;

adding watermark signals to said DCT values to obtain DCT values with watermark;

performing discrete cosine transformation on predicted images to obtain DCT values of predicted images;

subtracting DCT values of the predicted images from the DCT values with watermark to obtain DCT values of residual images;

quantizing the difference values to obtain quantized DCT values of residual images;

inverse quantizing the quantized DCT values at residual images and summing the inverse quantized DCT values of residual images with the DCT values of predicted images, for generating DCT coefficients for decoded images,

calculating correction signals from the DCT coefficients for decoded images;

summing the correction signals with the quantized DCT values of residual images to obtain an output signal;

inverse quantizing the output signal and inverse DCT the inverse quantized output signal and summing the resultant signal with predicted images to provide locally-decoded images;

storing the locally-decoded images in memory for generating the predicted images; and

valuable length encoding the output signal to provide a watermarked MPEG video signal of the input video images.

7. A method of inserting a watermark signal into video images as set forth in claim 6, where said calculating step generates a negative value as said correction signal when the extracted value exceeds the corresponding target value, and generates a positive value as said correction signal when the extracted value falls short of the corresponding target value.

8. A method of inserting a watermark signal into video images set forth in claim 7, where said positive value is +1 and said negative value is −1.

9. A method of inserting a watermark signal into video images as set forth in claim 7, where said calculating step generates a zero value as the correction signal if the corresponding quantized DCT values of residual images are zero.

10. A method of inserting a watermark signal into video images as set forth in claim 9, where said positive value is +1 and said negative value is −1.

* * * * *

US006104826A

United States Patent [19]

Nakagawa et al.

[11] Patent Number: 6,104,826

[45] Date of Patent: Aug. 15, 2000

[54] METHOD OF WATERMARK-EMBEDDING/ EXTRACTING IDENTIFICATION INFORMATION INTO/FROM PICTURE DATA AND APPARATUS THEREOF, AND COMPUTER READABLE MEDIUM

[75] Inventors: **Akira Nakagawa**; **Kimihiko Kazui**; **Atsuko Tada**; **Eishi Morimatsu**; **Koich Tanaka**, all of Kawasaki, Japan

[73] Assignee: **Fujitsu Limited**, Kawasaki, Japan

[21] Appl. No.: **08/948,083**

[22] Filed: **Oct. 9, 1997**

[30] **Foreign Application Priority Data**

Feb. 19, 1997 [JP] Japan 9-035258

[51] **Int. Cl.**^{7} **G06K 9/00**

[52] **U.S. Cl.** **382/100**; 382/248

[58] **Field of Search** 382/232, 100, 382/238; 380/4, 34, 55, 23, 54, 287, 210

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,848,155 12/1998 Cox 382/191
5,930,369 7/1999 Cox et al. 380/54

OTHER PUBLICATIONS

F.M. Boland J. J.K. O Rugnaidh and C. Dautzenberg "Watermarking Digital Images for Copyright Protection" Image Processing and its Applications Jul. 4–6 1995, Conference Publication No. 410 IEEE 1995 pp. 326–330, Apr. 1996.

Takeshi Ogihara et al. "Data Embedding into Pictorial Images with less Distortion Using Discrete Cosine Transform" Proceedings of ICPR '96 1015–4651/96 IEEE 1996 pp. 675–679.

Yasuhiro Nakamura et al. A Unified Cooling Method of Image and Text Data Using Discrete Cosine Transition Systems and Computers in Japan, vol. 21, No. 3, 1990 pp. 87–92, 1990.

Adriang Bors et al "Image Watermarking Using DCT Domain Constraint" IEEE 1996 Department of Informatics, University of Thessaloniki Greece, pp. 231–234, 1996.

I.J. Cox et al., "Secure Spread Spectrum Watermarking for Images, Audio and Video", 1996, pp. 243–246, Proceedings of the Int'l Conference of Image Processing, IEEE.

J.J.K.O. Ruanaidh et al., "Watermarking Digital Images for Copyright Protection", Aug. 1996, vol. 143, No. 4, pp. 250–256, IEE Proceedings: Vision, Image & Signal Processing, GB, Institute of Electrical Engineers.

Kineo Matsui et al., "Video–Stenography: How to Secretly Embed a Signature in a Picture", Jan. 1994, vol. 1, No. 1, Jan. 1994, pp. 187–206, IMA Intellectual Property Project Proceedins.

Jian Zhao et al., "Embedding Robust Labels Into Images for Copyright Protection", 1995, pp. 242–251, Proceedings of the Knowright. Conference. Proceedings of the Int'l Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technology.

Primary Examiner—Andrew W. Johns
Assistant Examiner—Shervin Nakhjavan
Attorney, Agent, or Firm—Staas & Halsey, LLP

[57] **ABSTRACT**

The present invention relates to methods for watermark-embedding/extracting identification information into/from picture data and apparatuses thereof. When identification information is watermark-embedded, the following steps are performed. Combinations of basis functions orthogonal each other are generated in association with each numerical signal. For each combination of the orthogonal basis functions, a weight coefficient is calculated so as to correspond to the combination of the basis functions by calculating a sum of products of values of the basis functions for pixels in the original picture data and pixel values of the pixels. For each numerical signal, a watermark-embedding function is referred, a input value is specified, and a pixel value is changed. When the identification information is extracted from the picture data, for each weight coefficient, the watermark-embedding function is referred, and the value of the watermark-embedding function is calculated for each weight coefficient.

17 Claims, 11 Drawing Sheets

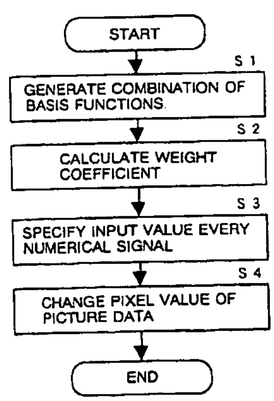


FIG. 1

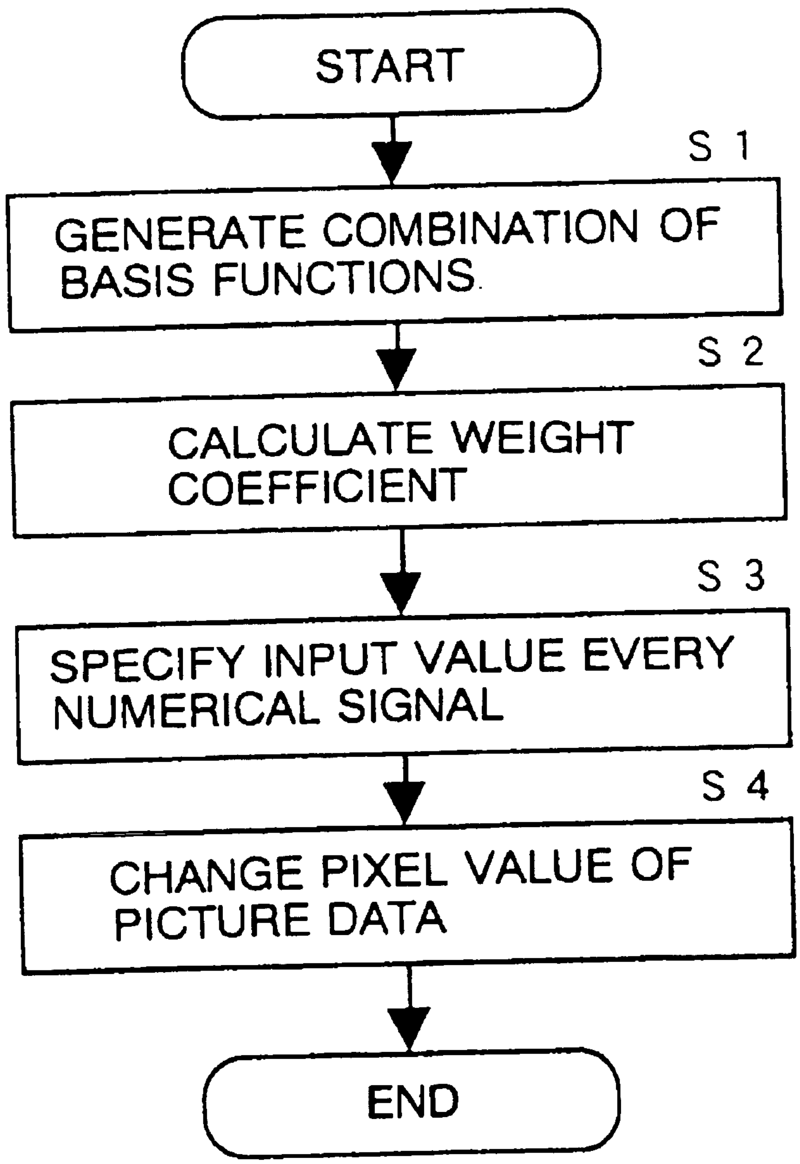

START
S 1
GENERATE COMBINATION OF BASIS FUNCTIONS
S 2
CALCULATE WEIGHT COEFFICIENT
S 3
SPECIFY INPUT VALUE EVERY NUMERICAL SIGNAL
S 4
CHANGE PIXEL VALUE OF PICTURE DATA
END

FIG. 2

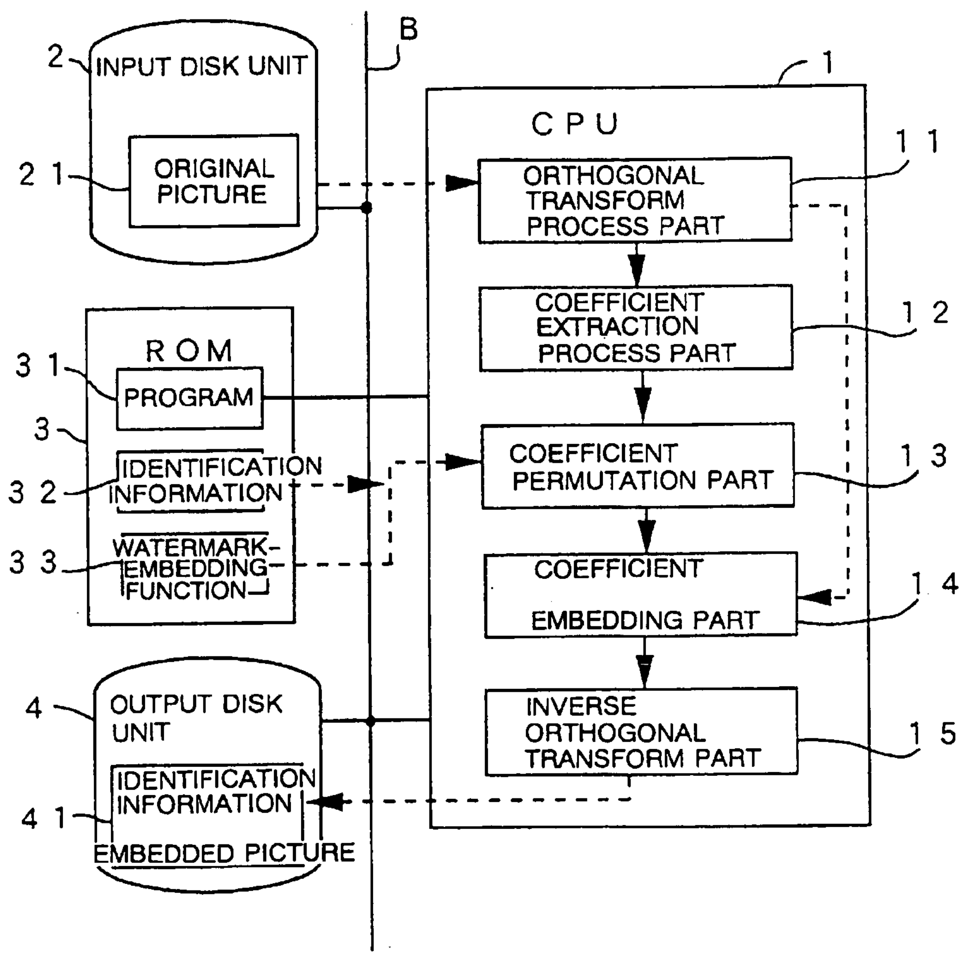

B
1
CPU
2
INPUT DISK UNIT
21
ORIGINAL PICTURE
11
ORTHOGONAL TRANSFORM PROCESS PART
COEFFICIENT EXTRACTION PROCESS PART
12
ROM
31
PROGRAM
3
32
IDENTIFICATION INFORMATION
COEFFICIENT PERMUTATION PART
13
33
WATERMARK-EMBEDDING-FUNCTION
COEFFICIENT EMBEDDING PART
14
4
OUTPUT DISK UNIT
INVERSE ORTHOGONAL TRANSFORM PART
15
41
IDENTIFICATION INFORMATION
EMBEDDED PICTURE

FIG. 3

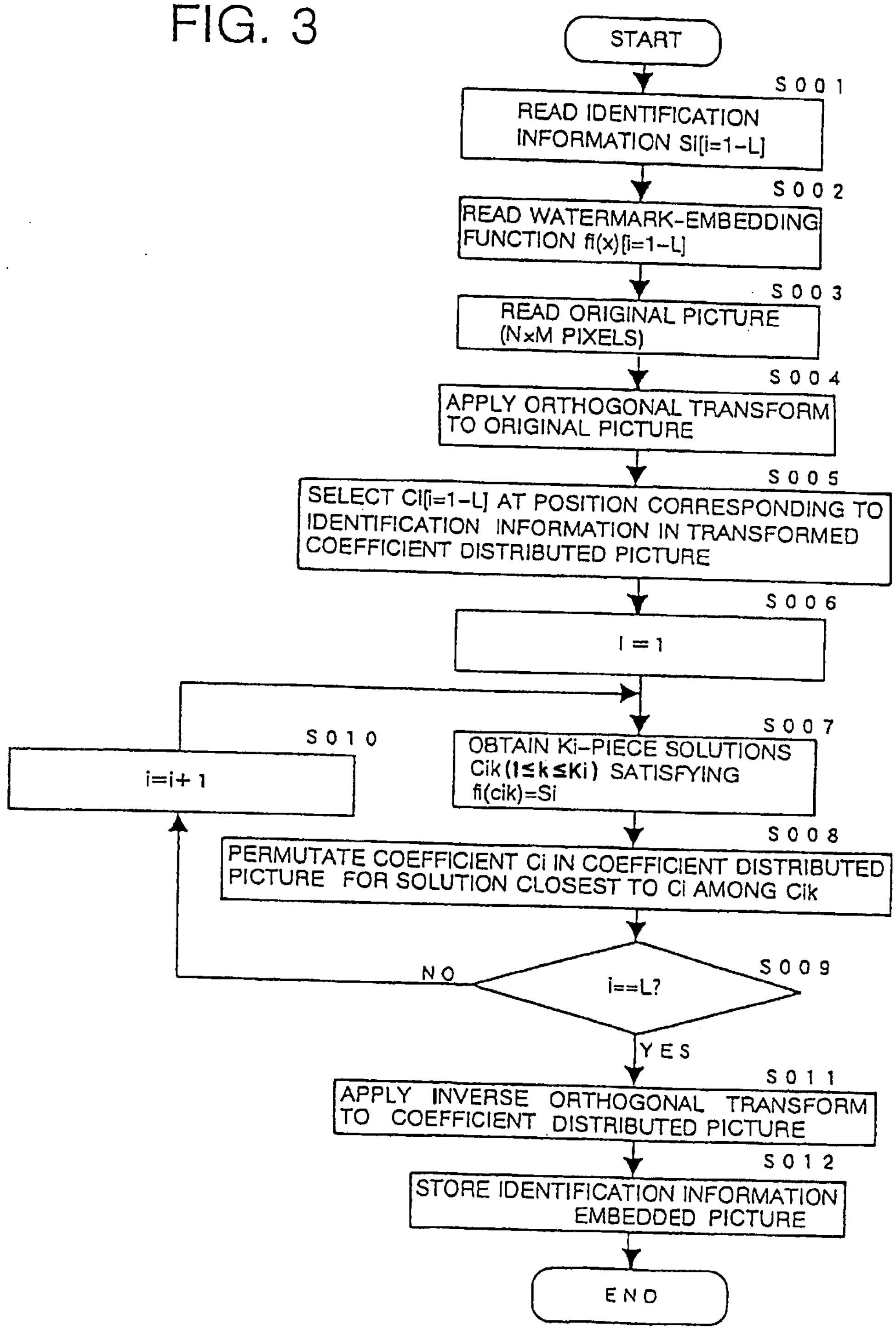

START
S001
READ IDENTIFICATION INFORMATION Si[i=1−L]
S002
READ WATERMARK-EMBEDDING FUNCTION fi(x)[i=1−L]
S003
READ ORIGINAL PICTURE (N×M PIXELS)
S004
APPLY ORTHOGONAL TRANSFORM TO ORIGINAL PICTURE
S005
SELECT Ci[i=1−L] AT POSITION CORRESPONDING TO IDENTIFICATION INFORMATION IN TRANSFORMED COEFFICIENT DISTRIBUTED PICTURE
S006
I = 1
S007
OBTAIN Ki-PIECE SOLUTIONS Cik(1≤k≤Ki) SATISFYING fi(cik)=Si
S010
i=i+1
S008
PERMUTATE COEFFICIENT Ci IN COEFFICIENT DISTRIBUTED PICTURE FOR SOLUTION CLOSEST TO Ci AMONG Cik
S009
i==L?
NO
YES
S011
APPLY INVERSE ORTHOGONAL TRANSFORM TO COEFFICIENT DISTRIBUTED PICTURE
S012
STORE IDENTIFICATION INFORMATION EMBEDDED PICTURE
END

FIG. 4

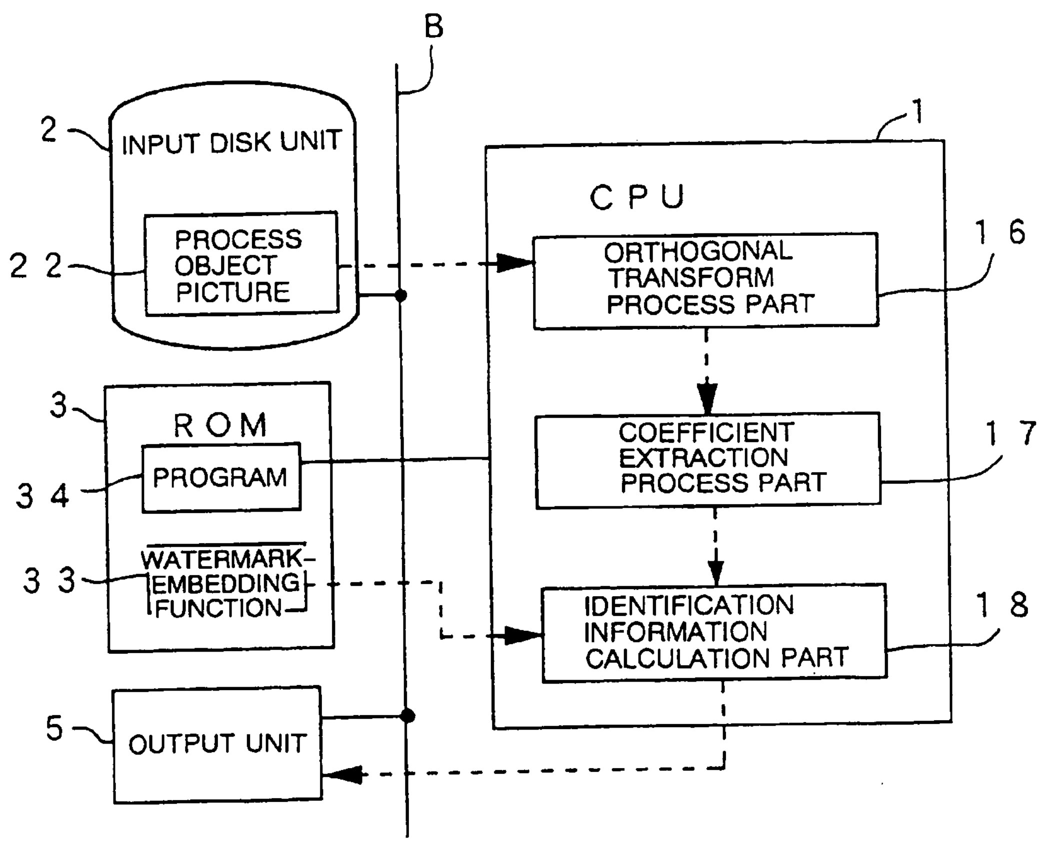

B
2
INPUT DISK UNIT
1
CPU
PROCESS OBJECT PICTURE
2 2
ORTHOGONAL TRANSFORM PROCESS PART
1 6
3
ROM
COEFFICIENT EXTRACTION PROCESS PART
1 7
PROGRAM
3 4
WATERMARK-EMBEDDING FUNCTION
3 3
IDENTIFICATION INFORMATION CALCULATION PART
1 8
5
OUTPUT UNIT

FIG. 5

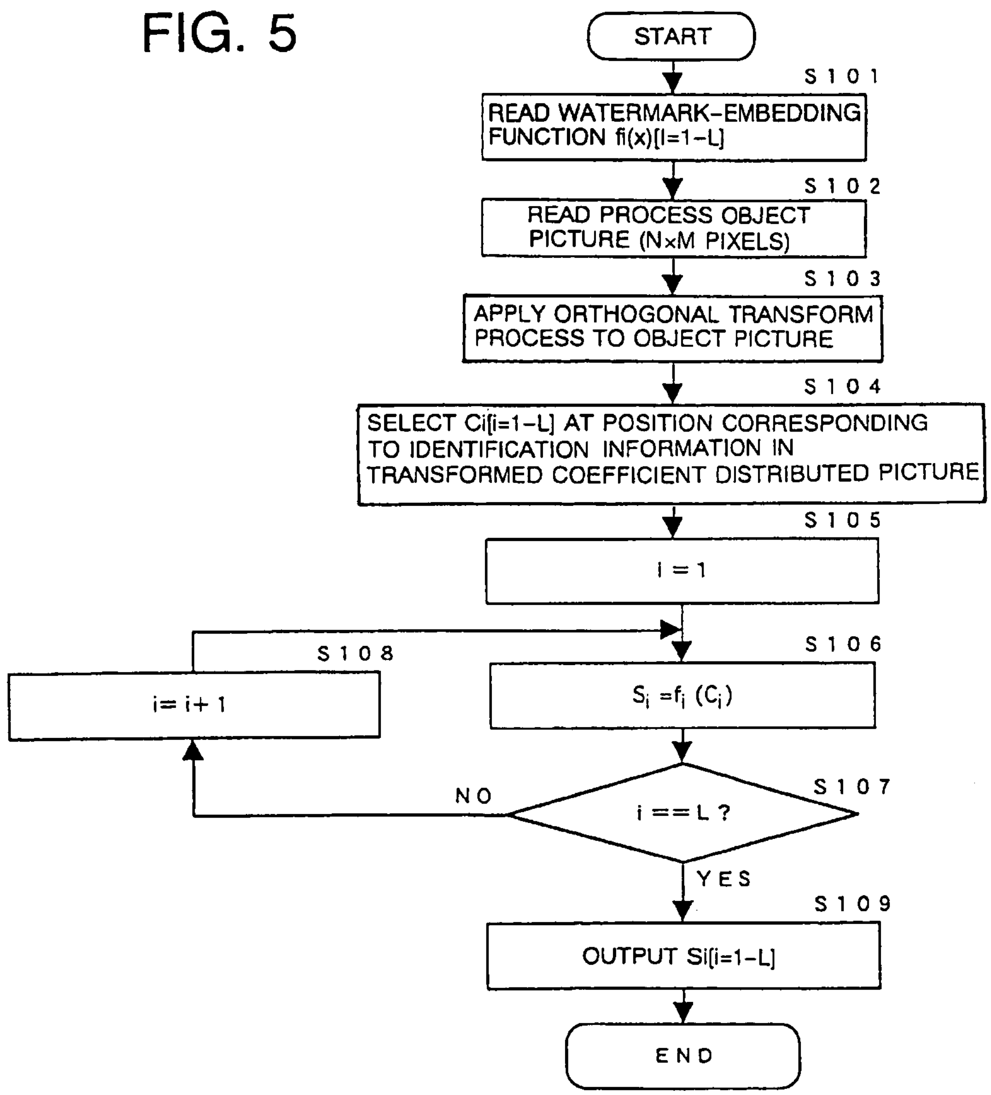
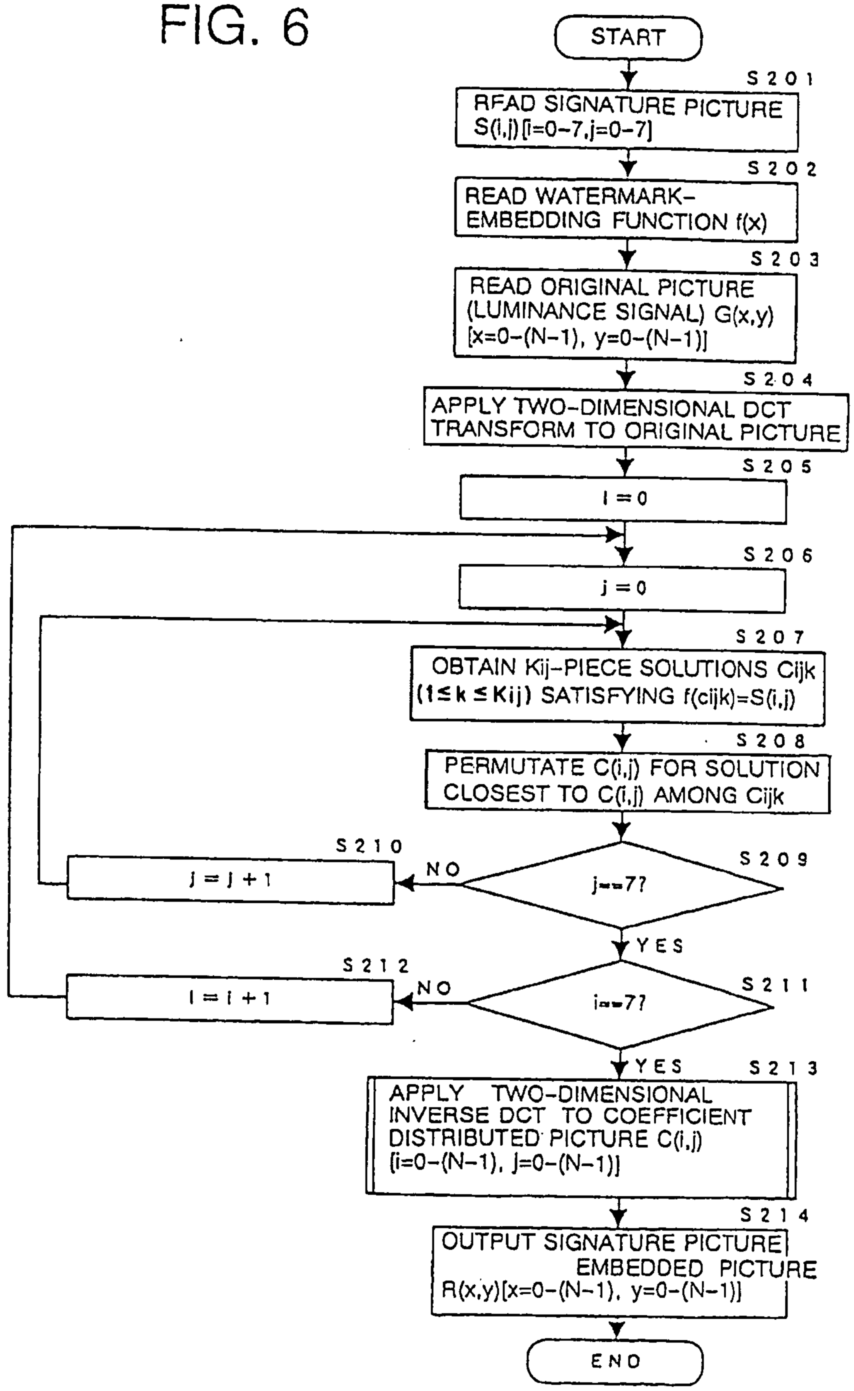

START
S101
READ WATERMARK-EMBEDDING FUNCTION fi(x)[I=1-L]
S102
READ PROCESS OBJECT PICTURE (N×M PIXELS)
S103
APPLY ORTHOGONAL TRANSFORM PROCESS TO OBJECT PICTURE
S104
SELECT Ci[i=1-L] AT POSITION CORRESPONDING TO IDENTIFICATION INFORMATION IN TRANSFORMED COEFFICIENT DISTRIBUTED PICTURE
S105
i = 1
S106
$S_i = f_i(C_i)$
S107
i == L ?
NO
YES
S108
i= i+ 1
S109
OUTPUT Si[i=1-L]
END

FIG. 6



START
S201
READ SIGNATURE PICTURE S(i,j) [i=0~7, j=0~7]
S202
READ WATERMARK-EMBEDDING FUNCTION f(x)
S203
READ ORIGINAL PICTURE (LUMINANCE SIGNAL) G(x,y) [x=0~(N-1), y=0~(N-1)]
S204
APPLY TWO-DIMENSIONAL DCT TRANSFORM TO ORIGINAL PICTURE
S205
i = 0
S206
j = 0
S207
OBTAIN Kij-PIECE SOLUTIONS Cijk (1≤k≤Kij) SATISFYING f(cijk)=S(i,j)
S208
PERMUTATE C(i,j) FOR SOLUTION CLOSEST TO C(i,j) AMONG Cijk
S209
j==7?
NO
S210
j = j + 1
YES
S211
i==7?
NO
S212
i = i + 1
YES
S213
APPLY TWO-DIMENSIONAL INVERSE DCT TO COEFFICIENT DISTRIBUTED PICTURE C(i,j) [i=0~(N-1), j=0~(N-1)]
S214
OUTPUT SIGNATURE PICTURE EMBEDDED PICTURE R(x,y) [x=0~(N-1), y=0~(N-1)]
END

FIG. 7

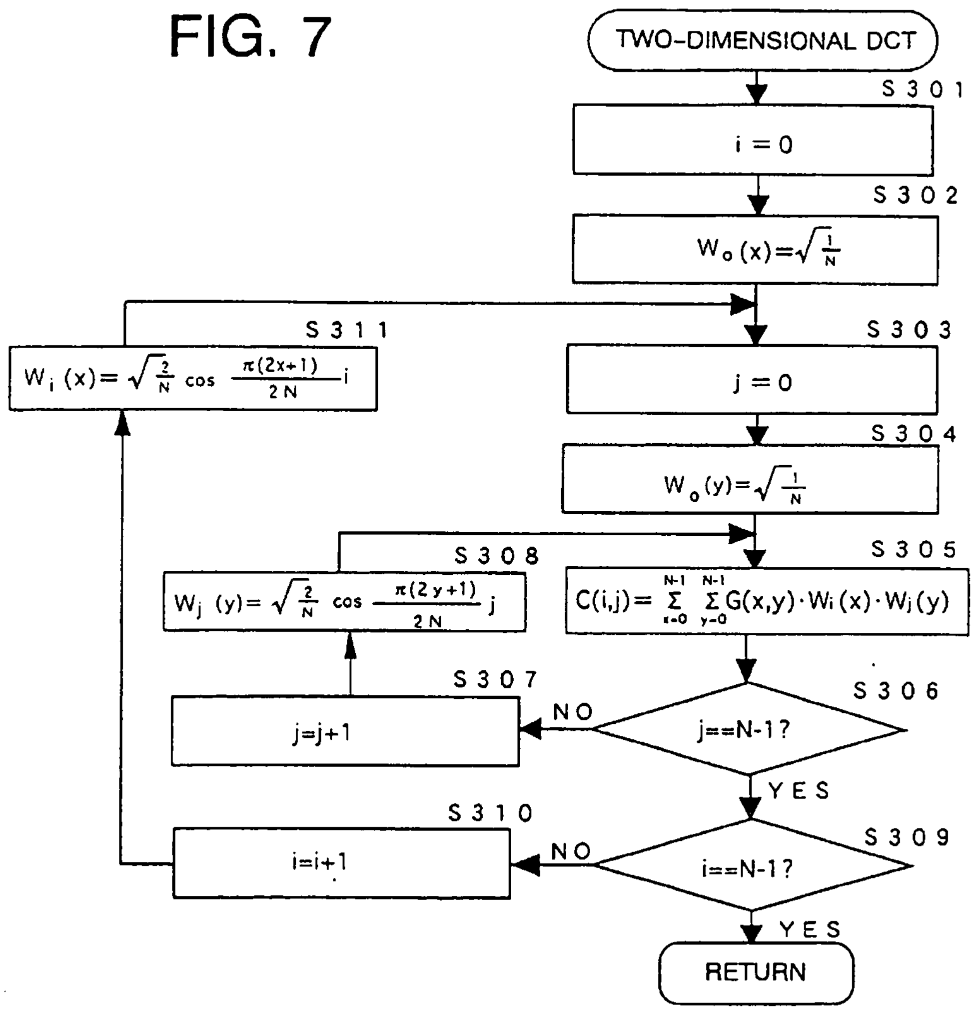

TWO-DIMENSIONAL DCT
S301
i = 0
S302
$W_0(x)=\sqrt{\frac{1}{N}}$
S303
j = 0
S304
$W_0(y)=\sqrt{\frac{1}{N}}$
S305
$C(i,j)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}G(x,y)\cdot W_i(x)\cdot W_j(y)$
S306
j==N-1?
NO
YES
S307
j=j+1
S308
$W_j(y)=\sqrt{\frac{2}{N}}\cos\frac{\pi(2y+1)}{2N}j$
S309
i==N-1?
NO
YES
S310
i=i+1
S311
$W_i(x)=\sqrt{\frac{2}{N}}\cos\frac{\pi(2x+1)}{2N}i$
RETURN

FIG. 8

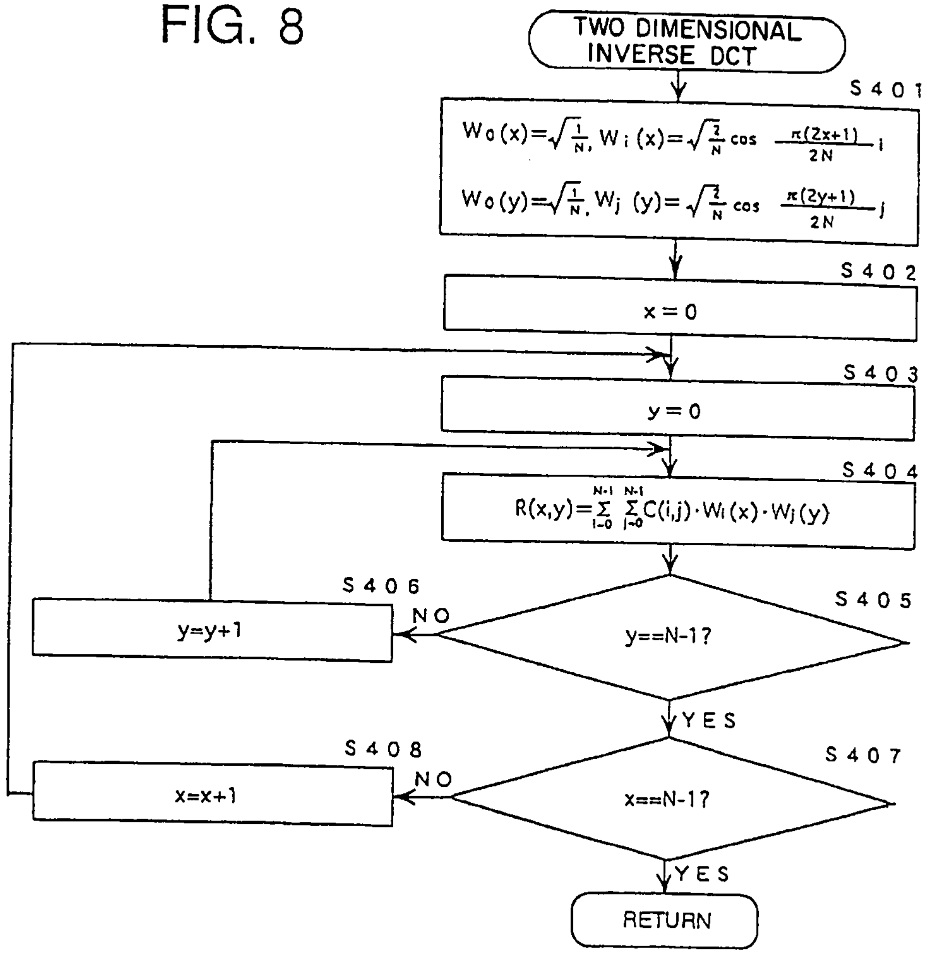
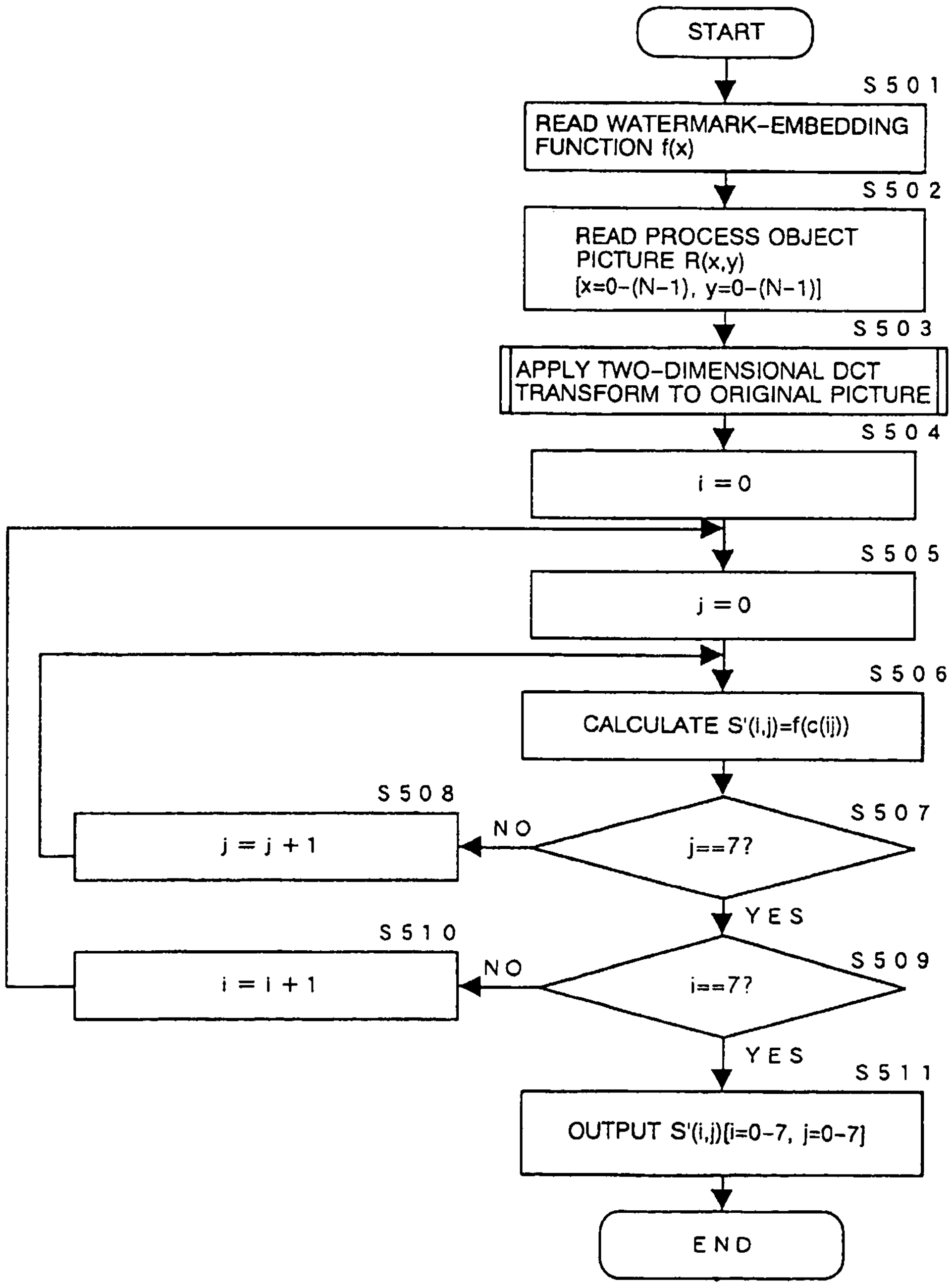

TWO DIMENSIONAL INVERSE DCT
S401
$W_0(x)=\sqrt{\frac{1}{N}}, W_i(x)=\sqrt{\frac{2}{N}}\cos\frac{\pi(2x+1)}{2N}i$
$W_0(y)=\sqrt{\frac{1}{N}}, W_j(y)=\sqrt{\frac{2}{N}}\cos\frac{\pi(2y+1)}{2N}j$
S402
x=0
S403
y=0
S404
$R(x,y)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}C(i,j)\cdot W_i(x)\cdot W_j(y)$
S405
y==N-1?
NO
S406
y=y+1
YES
S407
x==N-1?
NO
S408
x=x+1
YES
RETURN

FIG. 9


START
S501
READ WATERMARK-EMBEDDING FUNCTION f(x)
S502
READ PROCESS OBJECT PICTURE R(x,y) [x=0-(N-1), y=0-(N-1)]
S503
APPLY TWO-DIMENSIONAL DCT TRANSFORM TO ORIGINAL PICTURE
S504
i = 0
S505
j = 0
S506
CALCULATE S'(i,j)=f(c(ij))
S507
j==7?
NO
S508
j = j + 1
YES
S509
i==7?
NO
S510
i = i + 1
YES
S511
OUTPUT S'(i,j)[i=0-7, j=0-7]
END

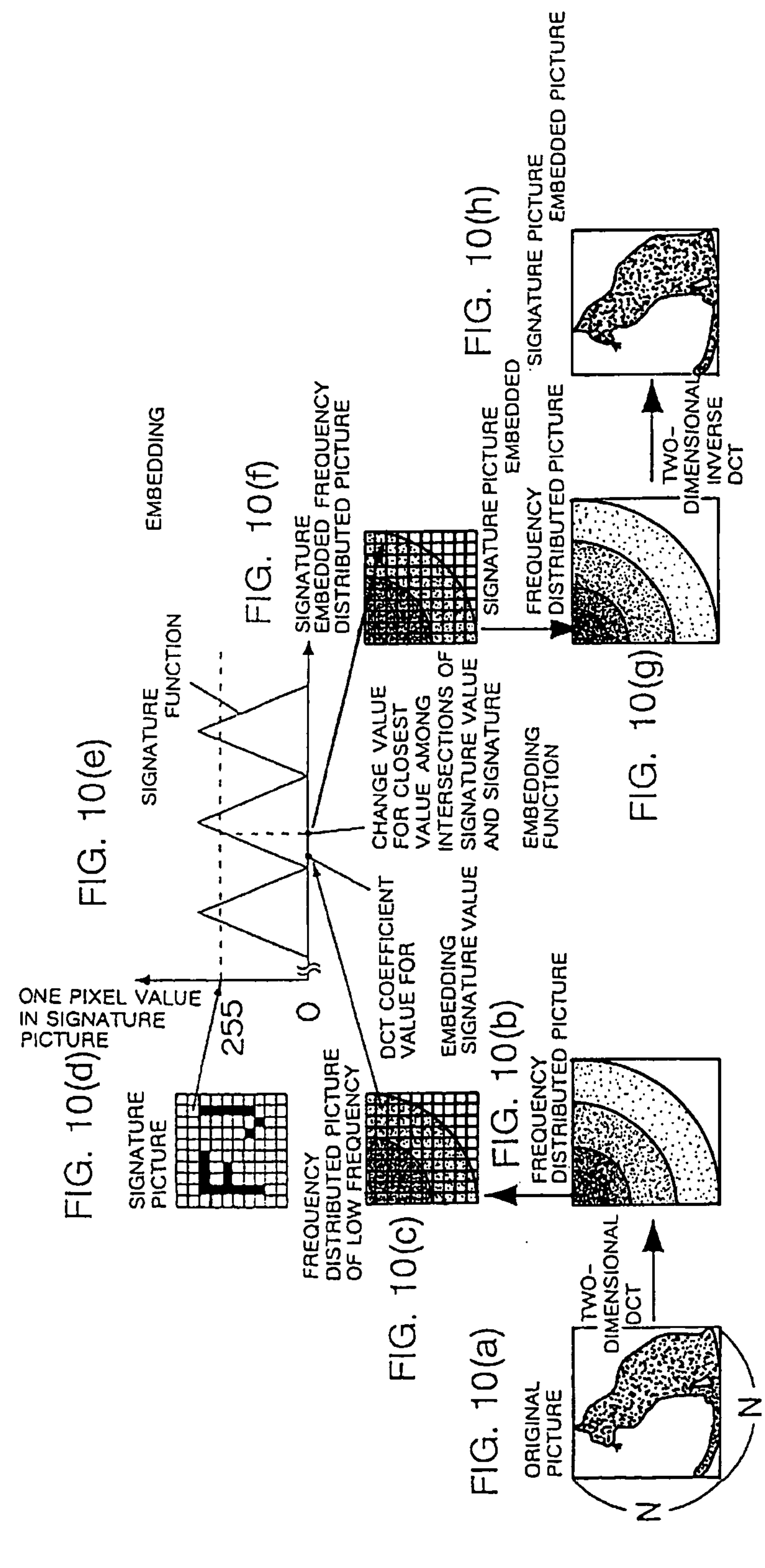


FIG. 10(a)
FIG. 10(b)
FIG. 10(c)
FIG. 10(d)
FIG. 10(e)
FIG. 10(f)
FIG. 10(g)
FIG. 10(h)

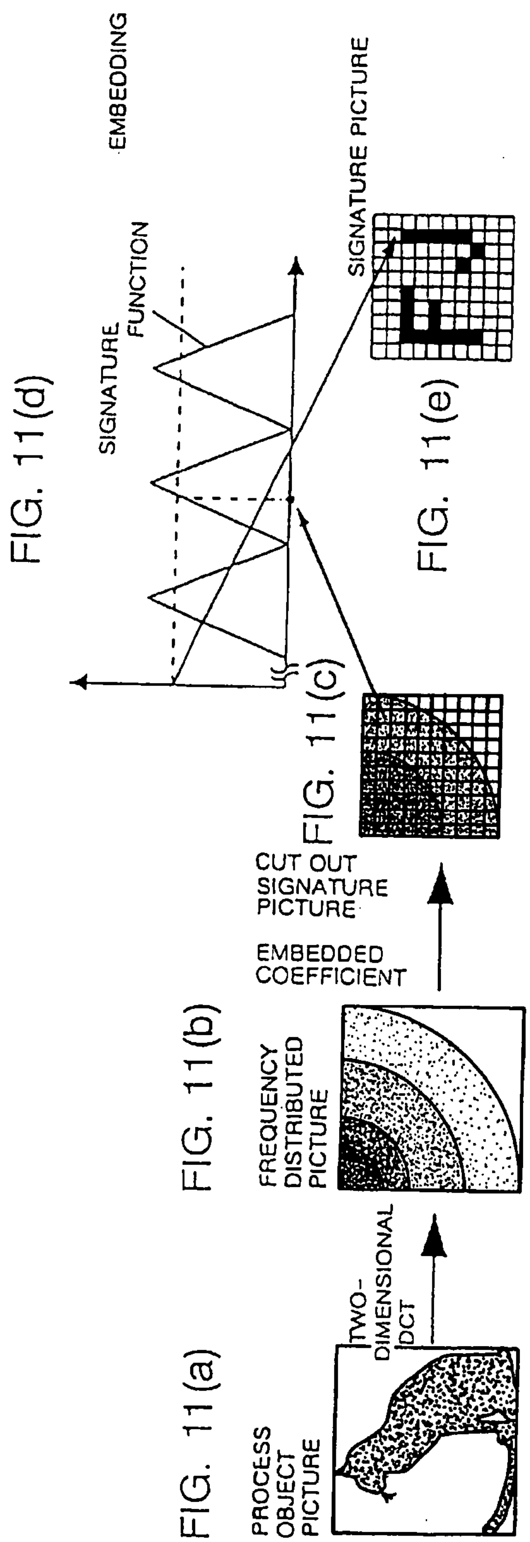


FIG. 11(a)

FIG. 11(b)

FIG. 11(c)

FIG. 11(d)

FIG. 11(e)

METHOD OF WATERMARK-EMBEDDING/ EXTRACTING IDENTIFICATION INFORMATION INTO/FROM PICTURE DATA AND APPARATUS THEREOF, AND COMPUTER READABLE MEDIUM

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to technology for watermark-embedding identification information indicating a person who has a right of copyright or the like into digital picture data distributed via various media such as circuits, and relates to technology for extracting identification information from those digital picture data.

2. Description of the Related Art

As digital technology and multi-media public advance in recent years, various data are converted into digital data and are distributed broadly via media such as a communication network, a satellite communication and a CD-ROM (Compact Disk Read Only Memory). When digital data are distributed in this multi-media public, it is not possible to avoid problems concerning copies of digital data.

When a copy of digital data is lawful, it may help the culture in the multi-media public. When, however, the copy is unlawful such as used directly for business, there is a possibility that the copy causes an enormous profit loss for a person who has a right (such as an author, a copyrighter, a copyright holder, a person who has neighboring right) since it is possible to copy digital data without deterioration.

Conventionally, technology is proposed in that identification information is watermark-embedded into picture data in a manner that it is difficult to recognize it by appearances, and the identification information is used as evidence when this picture data is copied unlawfully. For example, the following technology is proposed by J. Cox et al. in "Secure Spread Spectrum Watermarking for Multimedia", NEC Research Institute, Technical Report 95-10. That is, according to this technology, orthogonal transformation is applied to picture data, plural weight coefficients corresponding to dot positions of the identification information are selected among the weight coefficients of basis function obtained by the transformation, dot values of the identification information are respectively added to the selected weight coefficients, and the inverse orthogonal transformation is applied to all weight coefficients including the added coefficients, as the result, identification information watermark-embedded picture data is generated.

In the above-mentioned conventional technology, however, since the weight coefficients obtained by the orthogonal transformation of original picture data (before watermark-embedding identification-information) may be various values, it is not possible to specify weight coefficients to which dot values of the identification information are respectively added among the weight coefficients obtained by the orthogonal transformation of the identification information watermark-embedded picture data. Thus, to enforce the above-mentioned technology, it is necessary to keep and to manage each pair of original picture data and identification information watermark-embedded picture data, and when a copy appears, it is necessary to extract each dot value of identification information by subtracting each weight coefficient obtained by the orthogonal transformation of the kept and managed original data from each weight coefficient obtained by the orthogonal transformation of the kept and managed identification information watermark-embedded picture data and to extract each dot value of identification information by subtracting each weight coefficient obtained by the orthogonal transformation of the original picture data from each weight coefficient obtained by the orthogonal transformation of the copy, thereafter, it is necessary to prove the identify about both identification information.

According to the conventional technology, it is necessary to keep and to manage both identification information watermark-embedded picture data and original picture data twice, therefore, there are problems in that a data management and a proof to detect a copy are troublesome and twice storage are necessary compared with distributed data quantity. These problems are serious, particularly in a database in which data must be updated frequently, such as a database dealing picture data about 1000 pieces and a database for newspapers.

To avoid this twice data management, it is also considered that a part of the weight coefficient obtained by the orthogonal transformation of the original data is permutated by the just value of the identification information. There are possibilities in that this permutation causes a remarkable deterioration of the picture quality after the inverse orthogonal transformation, and in that the identification information is immediately recognized by a reproducer only by the orthogonal transformation of the picture data and then data is rewritten.

SUMMARY OF THE INVENTION

Accordingly, it is an object of the present invention to provide a method and an apparatus for watermark-embedding identification information into picture data, wherein identification information can be extracted from identification information watermark-embedded picture data without original picture data and identification information can be watermark-embedded into picture data without deteriorating the picture quality so as not to be recognized by a reproducer, to provide a method and an apparatus for extracting identification information from the picture data in which the identification information is watermark-embedded by those method and apparatus, and to provide a computer readable medium storing a program to make a computer function as an identification information watermark-embedding and/or extracting apparatus.

The preset invention is achieved to solve the above-mentioned problems.

That is, the first aspect of the present invention relates to an identification information watermark-embedding method for watermark-embedding identification information into original picture data consisting of pixel values which are arranged in a matrix. The numerical signals is not more than the pixel values. In the identification information watermark-embedding method, combinations of basis functions orthogonal each other are generated in association with each of the numerical signals (S1), weight coefficients are calculated so as that each of weight coefficients corresponds to each of the combinations of the basis functions by calculating, for each of the combinations of the basis functions, a sum of products, each of which are calculated for each of the pixels, based on a value of each of the basis functions for a position of the pixel within the original picture data and a pixel value of the pixel (S2), for each of the numerical signals, a watermark-embedding function which is a multi-to-one function taking available values of the weight coefficients in a domain and taking available values of the numerical signals in a range is referred, and a input value of the watermark-embedding function closest to

the weight coefficients corresponding calculated for the combinations of the basis functions associated with the numerical signal is specified among plural input values of the watermark-embedding function outputting a numerical value of the numerical signal (S3), pixel values in the original data are changed so that each of the weight coefficients becomes a value equal to the input value which is specified (S4).

According to the present invention, the numerical information of the identification information is not directly watermark-embedded into the weight coefficient, but the input value closest to the original weight coefficient is changed for the weight coefficient among the plural input values which the numerical information is the output value in the watermark-embedding function. Thus, the difference of the weight coefficient before and after changing becomes very small. As the result, there is little case that the picture quality of the picture data after changing the pixel value deteriorates. Further, since the weight coefficient after changing the pixel value corresponds to one numerical value of the numerical information to be identification information by the watermark-embedding function, the identification information can be extracted from the picture data after changing the picture value. Moreover, though third persons calculate the weight coefficient, they can not detect which value corresponds to each weight coefficient without the watermark-embedding function. Thus, since the third persons can not know contents of the identification information, they can not modify the identification information.

The pixel value in the original picture data may be a luminance value of each color signal in RGB signal, a luminance value in YCC signal or a color difference value in YCC signal.

The dot number in the original picture data may be set so as to be equal or different lengthwise and breadthwise.

The available numerical value of each numerical value signal in the identification information may be binary or more than.

The numerical signals in the identification information may be arranged in a line or may be picture information arranged in a matrix.

The weight coefficient may be calculated only for a combination of basis functions corresponding to each numerical signal in the identification information, or may be calculated for all combinations of basis functions by the orthogonal transformation. According to the former, the total of the calculation process for the weight coefficient reduces. According to the latter, it is possible to change the pixel value of the picture data only by applying the inverse orthogonal transformation to all weight coefficient after permutating the weight coefficient for the combination of the basis functions corresponding to each numerical signal in the identification information. As this orthogonal transformation, a two-dimensional discrete cosine transformation, a two-dimensional discrete sine transformation, or a Hadamard transformation may be used.

The watermark-embedding function may be held by functional expressions or by tables providing relations of one output value for plural input values. The watermark-embedding function may be a periodic function or not, as long as a multi-one function is used. When a periodic function is used, functional expressions become simple, therefore, it is possible to simplify an apparatus to carry out the present invention. That is, only if a intersection with a value of each numerical signal of the identification information in one period of the watermark-embedding function is obtained, it becomes easy to specify input values because there are solutions at points that the intersection is shifted by times of whole numbers. Additionally, from points of the picture quality in the picture data after changing the pixel values, it becomes subjectively difficult to watch errors by this change for components of which weight coefficients are larger. The change degrees are larger, resistance for various picture data change becomes stronger. To make the change degree large, a period of the watermark-embedding function may be made long. Thus, a watermark-embedding function may be used in which the period becomes long in accordance with that the value of the weight coefficient becomes large in a manner that the weight coefficient is changed to a larger value for a larger weight coefficient, and the weight coefficient is changed to a small value for a smaller weight coefficient.

The second aspect of the present invention relates to an identification information extracting method for extracting identification information from process object picture data into which identification information is watermark-embedded by the identification information watermark-embedding method according to the first aspect. In the identification information extracting method, combinations of basis functions orthogonal each other is generated in association with each of the numerical signals. For each of the combinations of the orthogonal basis functions, weight coefficients are calculated so that each of weight coefficients corresponds to each of the combinations of the basis functions by calculating a sum of products, each of which are calculated, for each of the pixels, based on a value of each of the basis functions for a position of the pixel in the original picture data and a pixel value of the pixel. Then, for each of the weight coefficients, a watermark-embedding function which is a multi-to-one function taking available values of the weight coefficients in a domain and taking available values of the numerical signals in a range is referred and a value of the watermark-embedding function for each of the weight coefficients is calculated.

The third aspect of the present invention relates to an identification information watermark-embedding method for watermark-embedding identification information into original picture data consisting of pixel values which are arranged in a matrix. The identification information consists of numerical signals not more than the pixel values. In the identification information watermark-embedding method, orthogonal transformation is applied to each of pixel values, and coefficient distributed data consisting of weight coefficients which are arranged in a matrix is generated, each of weight coefficients selected from the coefficient distributed data is related with one of the numerical signals. For each of the numerical signals, a watermark-embedding function which is a multi-to-one function taking available values of the weight coefficients in a domain and taking available values of the numerical signals in a range is referred, an input value of the watermark-embedding function to the weight coefficients related with the numerical signal is specified among plural input values of which output values correspond to a numerical value of one of the numerical signal, and the weight coefficients in the coefficient distributed data related with the numerical signal is permutated by the input value. Then, inverse orthogonal transformation is applied to the coefficient distributed data after permutating the weight coefficients for all of the numerical signals.

The fourth aspect of the present invention relates to an identification information extracting method for extracting identification information from process object picture data to which the identification information is watermark-

embedded by the identification information watermark-embedding method according to the third aspect. In the identification information extracting method, orthogonal transformation is applied to each pixel values and coefficient distributed data consisting of weight coefficients which are arranged is generated in a matrix, weight coefficients corresponding to each of the numerical signals are picked up from the coefficient distributed data. For each of the weight coefficients which are picked up, a watermark-embedding function which is a multi-to-one function taking available values of the pickup weight coefficients in a domain and taking available values of the numerical signals in a range is referred, and a value of the watermark-embedding function for the each of the weight coefficients which are picked up is calculated.

The fifth aspect of the present invention relates to an identification information watermark-embedding method for watermark-embedding identification information into original picture data consisting of pixel values which are arranged in a matrix. The identification information consists of numerical signals not more than the pixel values. In the identification information watermark-embedding method, two-dimensional discrete cosine transformation is applied to each of pixel values, and generating coefficient distributed data consisting of weight coefficients which are arranged in a matrix, each of weight coefficients selected from the coefficient distributed data is related with one of the numerical signals. For each of the numerical signals, a watermark-embedding function which is a multi-to-one function taking available values of the selected weight coefficients in a domain and taking available values of the numerical signals in a range is referred, an input value closest to the weight coefficients related with the numerical signals is specified among plural input values which output values correspond to a numerical value of the numerical signal, and the weight coefficients in the coefficient distributed data related with the numerical signal is permutated by the closest input value. Then, inverse two-dimensional discrete cosine transformation is applied to the coefficient distributed data after permutating the weight coefficients for all of the numerical signals.

The sixth aspect of the present invention relates to an identification information extracting method for extracting identification information from process object picture data into which the identification information is watermark-embedded by the identification information watermark-embedding method according to the fifth aspect. In the identification information extracting method, two-dimensional discrete cosine transformation is applied to each pixel values, and coefficient distributed data consisting of weight coefficients which are arranged in a matrix is generated. Weight coefficients corresponding to the numerical signals are picked up from the coefficient distributed data. For each of the weight coefficients which are picked up, a watermark-embedding function which is a multi-to-one function taking available values of the weight coefficients in a domain and taking available values of the numerical signals in a range is referred, and a value of the watermark-embedding function for the each of the weight coefficients which are picked up is calculated.

The seventh aspect of the present invention relates to an identification information watermark-embedding method according to the first, third, or fifth aspect, and is specified by that the watermark-embedding function is a period function.

The eighth aspect of the present invention relates to an identification information extracting method according to the second, fourth, or sixth aspect, and is specified by that the watermark-embedding function is a period function.

The ninth aspect of the present invention relates to an identification information watermark-embedding method according to the first, third, or fifth aspect, and is specified by that the watermark-embedding function is a continuous period function.

The tenth aspect of the present invention relates to an identification information extracting method according to the second, fourth, or sixth aspect, and is specified by that the watermark-embedding function is a continuous period function.

The eleventh aspect of the present invention relates to an identification information watermark-embedding method according to the first, third, or fifth aspect, and is specified by that the watermark-embedding function makes an interval between the plural input values taking one output value narrow when an input value is low, and makes the interval broad when the input value is high.

The twelfth aspect of the present invention relates to an identification information extracting method according to the second, fourth, or sixth aspect, and is specified by that the watermark-embedding function makes an interval between the plural input values taking one output value narrow when an input value is low, and makes the interval broad when the input value is high.

The thirteenth aspect of the present invention relates to an identification information watermark-embedding apparatus of watermark-embedding identification information into original picture data consisting of pixel values which are arranged in a matrix. The identification information consists of numerical signals not more than the pixel vales. The identification information watermark-embedding apparatus comprises a watermark-embedding function hold unit for holding a watermark-embedding function which is a multi-to-one function taking available values of the weight coefficients in a domain and taking available values of the numerical signals in a range, an orthogonal transformation unit for applying orthogonal transformation to each of pixel values, and generating coefficient distributed data consisting of weight coefficients which are arranged in a matrix, a weight coefficient permutation unit for relating each of weight coefficients selected from the coefficient distributed data with one of the numerical signals, specifying an input value closest to the weight coefficients related with the numerical signals among plural input values of the watermark-embedding function outputting a numerical value of the numerical signal, and permutating the weight coefficient in the coefficient distributed data by the closest input value, and an inverse orthogonal transformation unit for applying inverse orthogonal transformation to the coefficient distributed data which the weight coefficients are permutated.

The fourteenth aspect of the present invention relates to an identification information extracting apparatus for extracting identification information from process object picture data into which the identification information is watermark-embedded by the identification information watermark-embedding apparatus according to the thirteenth aspect. The identification information extracting apparatus comprises a watermark-embedding function hold unit for holding a watermark-embedding function which is a multi-to-one function taking available values of the weight coefficients in a domain and taking available values of the numerical signals in a range, an orthogonal transformation unit for applying orthogonal transformation to each of pixel

values and generating coefficient distributed data consisting of weight coefficients which are arranged in a matrix, a pickup unit for picking up weight coefficients corresponding to each of the numerical signals from the coefficient distributed data, and a calculation unit for calculating a value of the watermark-embedding function for each of the weight coefficients which are picked up.

The fifteenth aspect of the present invention relates to a computer readable medium storing a program to control a computer. The program performs a step of applying orthogonal transform to each of pixel values and generating coefficient distributed data consisting of weight coefficients which are arranged in a matrix; a step of relating each of weight coefficients selected from the coefficient distributed data with the numerical signals; a step of, for each of the numerical signals, referring to a watermark-embedding function which is a multi-to-one function taking available values of the weight coefficients in a domain and taking available values of the numerical signals in a range, specifying an input value closest to weight coefficients related with the numerical signals among plural input values of the watermark-embedding function outputting a numerical value of the numerical signal, and permutating the weight coefficients in the coefficient distributed data by the closest input value; and a step of applying inverse orthogonal transformation to the coefficient distributed data after permutating the weight coefficients for all of the numerical signals.

The sixteenth aspect of the present invention relates to a computer readable medium storing a program to control a computer. The program performs a step of applying orthogonal transform to each pixel values, and generating coefficient distributed data consisting of weight coefficients which are arranged in a matrix; a step of picking up weight coefficients corresponding to the numerical signals from the coefficient distributed data; and a step of, for each of the weight coefficients which are picked up, referring to a watermark-embedding function which is a multi-to-one function taking available values of the pickup weight coefficients in a domain and taking available values of the numerical signals in a range and calculating a value of the watermark-embedding function for the each of the pickup weight coefficients.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and advantages of the present invention will become apparent during the following discussion conjunction with the accompanying drawings, in which:

FIG. **1** is a view illustrating a principal of the present invention;

FIG. **2** is a block diagram illustrating an outline structure of a watermark-embedding computer according to the aspect of the present invention;

FIG. **3** is a flow chart illustrating an identification information watermark-embedding process executed by the CPU **1** in FIG. **2**;

FIG. **4** is a block diagram illustrating an outline structure of an extracting computer according to the aspect of the present invention;

FIG. **5** is a flow chart illustrating an identification information extracting process performed by the CPU **1** in FIG. **4**;

FIG. **6** is a flow chart illustrating an identification information watermark-embedding process in Embodiment 1;

FIG. **7** is a flow chart illustrating a two-dimensional DCT process subroutine executed in **S204** in FIG. **6**;

FIG. **8** is a flow chart illustrating a two-dimensional DCT process subroutine executed in **S213** in FIG. **6**;

FIG. **9** is a flow chart illustrating an identification information extracting process in the Embodiment;

FIGS. **10**(*a*) through **10**(*h*) are explanatory views illustrating a flow of a signature picture watermark-embedding process in the Embodiment; and

FIG. **11**(*a*) through **11**(*e*) are explanatory views illustrating a flow of a signature picture extracting process in the Embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred aspects and embodiments of the present invention will now be described with reference to the accompanying drawings.

A watermark-embedding computer according to an aspect of a method and an apparatus for watermark-embedding identification information into image data and an extracting computer according to an aspect of a method and an apparatus for extracting identification information from image data according to the present invention are structured in a manner that conventional various orthogonal transformation processes and inverse orthogonal transformation processes can be used, and any original picture and identification information of various sizes can be processed (only, identification information≦original picture).

[Outline of Identification Information Watermark-embedding and Extracting]

First, explanations are given of outlines of watermark-embedding identification information into picture data and of extracting identification information from into identification information watermark-embedded picture data before explaining a concrete structure in this aspect.

Now, it is assumed that original picture data (such as monochrome picture data or luminance data extracted from NTSC color picture data) to be an identification information watermark-embedded object is structured by N×N pixels. It is also assumed that the identification information to be a watermark object is data including L (L≦N×M) signals to which predetermined values are respectively given.

The watermark-embedding computer applies the orthogonal transformation to all of the above-mentioned original picture data or to each of plural blocks obtained by dividing the original picture data (only, not less than L pixels in each block). The original picture is divided and the orthogonal transformation is applied to those, because a signature picture can be extracted without deterioration though a picture to which a watermark-embedding process is applied is cut off partially and copied. Incidentally, when a size of each partial picture is made too small, the picture to which the watermark-embedding process is applied deteriorates, therefore, it is preferable that each partial picture is not less than 8x8 pixels.

For the orthogonal transformation, L combinations of basis functions which are orthogonal one another in relation to each signal of the identification information are prepared. Each weight coefficient of combinations of basis functions is obtained by calculating a sum of products of a value of each basis function orthogonal to a position of each pixel in the original picture data and a luminance value thereof for each combination of basis functions. Incidentally, "orthogonal" indicates that directions of variables of each basis function are orthogonal one another in an original picture or a block including N×M pixels.

Then, the watermark-embedding computer watermarks each value of the signal included in the identification infor-

mation into each weight coefficient for the combination of basis functions which are previously related. At that time, the watermark-embedding computer does not directly add/write a value of each signal to/over a weight coefficient, but prepares a predetermined watermark-embedding function every signal and permutates a value corresponding to each signal for an original weight coefficient by this watermark-embedding function.

The watermark-embedding function prepared every signal may be defined so as to be different every signal or may be defined in common for signals. In any case, each watermark-embedding function includes a value of a weight coefficient within a defined range, and includes a value of each signal in the identification information within a numerical range. Each watermark-embedding function is a multi-one function in which plural input values in the defined range correspond to the same output value. Concretely, a process becomes simple when a simple function is used, therefore, it is desirable to use a periodic function (a continuous periodic function in which an interval between plural input values to be the same output value is narrow when the input value is small and the interval is broad when the input value is large) indicated by the following expression (1) and so on.

$$f(x) = A \cdot \sin\left(\frac{x}{T}\right) \tag{1}$$

The watermark-embedding computer, every signal in the identification information, inverse-calculates all input values in the watermark-embedding function in which a value of that signal is taken for a output value. Then, the watermark-embedding computer specifies an input value which has the smallest difference from the weight coefficient of the combination of the basis functions previously corresponded to the signal among all calculated input values, and permutates the specified input value for the original weight coefficient value.

Thereafter, the watermark-embedding computer rewrites a value of each pixel in the original picture data or in the block in a manner that each original weight coefficient of the combination of basis functions after the orthogonal transformation becomes a permutated value. Identification information watermark-embedded picture data is obtained in this way. As above mentioned, since the variation by permutating the weight coefficient is limited to the minimum, the deterioration of the identification information watermark-embedded picture data is kept to the minimum. Further, although a third person applies the orthogonal transformation to the identification information watermark-embedded picture data, the person can not know each signal value of the identification information in a case that the watermark-embedding function is not known, therefore, it is not possible to change the identification information.

The extracting computer applies the orthogonal transformation to all of process object picture data or to each of plural blocks obtained by dividing the process object picture data (only, not less than L pixels in each block). At that time, as well as for the case that the identification information is watermark-embedded into the original picture data, L combinations of basis functions which are orthogonal to one another in relation to each signal of the identification information are available. Each weight coefficient of the combination of basis functions is obtained by calculating a sum of products of respective values of basis functions orthogonal to positions of pixels in the process object picture data and luminance values thereof for combinations of basis functions. Incidentally, as basis functions, the same functions for watermark-embedding the identification information into the original picture data are used.

The extracting computer inputs each obtained weight coefficient into the corresponding watermark-embedding function so as to obtain an output value. Then, each obtained output value of the watermark-embedding function is arranged in accordance with the order of the signals in the identification information which are previously related with the corresponding weight coefficients. Then, when the process object picture data is the identification information watermark-embedded picture data, the arranged output values coincide with the identification information. In this way, the extracting computer can extract the identification information from the identification information watermark-embedded picture data without original picture data only with each watermark-embedding function. The watermark-embedding function may be used for various original picture data in common. Thus, the total of data to be kept and managed reduces exceedingly compared with the conventional approach.

[Structure of Watermark-embedding Computer]

Next, an explanation is given of a concrete structure of the watermark-embedding computer. FIG. 2 is an outline block diagram illustrating only a structure relative to a process for watermark-embedding identification information into original picture data in hardware of this watermark-embedding computer. As shown in FIG. 2, the watermark-embedding computer is provided with a CPU (Central Processing Unit) **1**, an input disk unit **2**, a ROM (Read Only Memory) **3** and an output disk unit **4** which are connected one another by a bus B.

The input disk unit **2** inputs original picture data **21** into the CPU **1** in accordance with an instruction from the CPU **1**, such as a hard disk unit, a floppy disk unit, or an optical magnetic disk unit.

The ROM **3** used as a watermark-embedding function hold unit and a computer readable medium is a read only memory holding an identification information watermark-embedding program **31** executed in the CPU **1**, identification information **32** and a watermark-embedding function **33**.

The CPU **1** is a processor controlling all of the watermark-embedding computer. The identification information watermark-embedding program **31** read from the ROM **3** runs, whereby an orthogonal transformation process part **11**, a coefficient extraction process part **12**, a coefficient permutation part **13**, a coefficient watermark-embedding part **14** and an inverse orthogonal transformation part **15** are implemented in the CPU **1**, and the identification information watermark-embedding process shown in FIG. **3** is executed. Dot lines in FIG. **2** indicate data flows in the CPU **1**.

The orthogonal transformation process part **11** used as an orthogonal transformation unit applies the orthogonal transformation to the original picture data **21** of N×M pixels read from the input disk unit **2**, and then calculates a weight coefficient for each of N×M combinations of the basis functions. The N×M weight coefficients forms a N×M matrix similarly to the original picture. The matrix of those weight coefficients, hereinafter, is called "coefficient distributed picture data" for convenience. The orthogonal transformation process part **11** informs both the coefficient extraction process part **12** and the coefficient watermark-embedding part **14** of this coefficient distributed picture data.

The coefficient extraction process part **12** extracts L weight coefficients to which signals of identification information are respectively watermark-embedded from the coefficient distributed picture data informed from the orthogonal

transformation process part **11**, and informs the coefficient permutation part **13** of them.

The coefficient permutation part **13** used as a coefficient permutation unit reads the identification information **32** and the watermark-embedding function **33** from the ROM **3**. Then, the watermark-embedding function **33** is inverse-calculated every signal of the identification information **32**, and then all input values, which take the output value for the value of the signal, of the watermark-embedding function **33** are obtained. Then, the coefficient permutation part **13** specifies an input value which has the smallest difference from the corresponding weight coefficient informed from the coefficient extraction process part **12** among the obtained input values, and informs the coefficient watermark-embedding part **13** of the specified input value as a rewrite value for that coefficient.

The coefficient watermark-embedding part **14** used as a coefficient watermark-embedding unit writes this rewrite value over the weight coefficient corresponding to the rewrite value informed from the coefficient watermark-embedding part **13** in the coefficient distributed picture data received from the orthogonal transformation process part **11**.

The inverse orthogonal transformation part **15** used as an inverse orthogonal transformation unit applies the inverse orthogonal transformation to the N×M weight coefficients received from the coefficient watermark-embedding part **14**. The inverse orthogonal transformation part **15**, every pixel in the identification information watermark-embedded picture, calculates a total of products of each weight coefficient corresponding to each combination of the basis functions used in the orthogonal transformation part **11** and each value of basis function corresponding to the position of that pixel, and then obtains a luminance value of that pixel. The inverse orthogonal transformation part **15** sends an identification information watermark-embedded picture data **41** including each pixel of which the luminance value is obtained like this to the output disk unit **4**.

In the output disk unit **4**, such as a hard disk unit, a floppy disk unit and an optical magnetic disk unit, the identification information watermark-embedded picture data **41** received from the CPU **1** is written.

FIG. **3** is a flow chart illustrating an identification information watermark-embedding process performed by the CPU **1** which has read the identification information watermark-embedding program **31**. The identification information watermark-embedding process starts by inputting an identification information watermark-embedding command via a keyboard (not shown) connected to the watermark-embedding computer.

In **S001** executed initially after starting the identification information watermark-embedding process, the CPU **1** reads L signals S_i [i=1–L] to be in the identification information **32** from the ROM **3**.

In **S002**, the CPU **1** reads L watermark-embedding functions **33** ($f_i(x)$[i=1–L]) from the ROM **3**.

In **S003**, the CPU **1** reads original picture data of N×M pixels from the input disk unit **2**.

In **S004**, the CPU **1** applies the above mentioned orthogonal transformation process to all of the original picture data read in **S003** so as to generate coefficient distributed picture data of N×M dots.

In **S005**, the CPU **1** selects a weight coefficient C_i [i=1–L] corresponding to each signal S_i[i=1–L] read in **S001** among weight coefficients included in the coefficient distributed picture data generated in **S004**.

In **S006**, the CPU **1** initializes a variable i to specify the weight coefficient C_i to be a process object to "1".

Next, the CPU **1** executes a loop process between **S007** and **S010**. In **S007** to be the first step of this loop, the CPU **1** obtains K_i-piece solutions c_{ik} ($1<k<K_i$) which satisfy the relation $f_i(c_{ik})=S_i$ based on the identification information signal S_i corresponding to the weight coefficient C of the process object and the watermark-embedding function $f_i(x)$. In this case, it is defined that a number of solutions c_{ik} for the watermark-embedding function $f_i(x)=S_i$ prepared correspondingly to the ith signal S_i is K_i.

In **S008**, the CPU **1** rewrites a weight coefficient C_i of the process object in the coefficient distributed picture data generated in **S004** by a solution closest to the weight coefficient C_i among solutions c_{ik} obtained in **S007**.

In **S009**, the CPU **1** checks whether the variable i gets to L or not, namely, whether processes in **S007** and **S008** are applied to all signals included in the identification information. Then, when the-variable i does not get to L, the CPU **1** increases the variable i, and then returns the operation to the **S007**.

On the contrary, when the variable i gets to L, the CPU **1** applies the above-mentioned inverse orthogonal transformation to the coefficient distributed picture rewritten in **S008** so as to generate identification information watermark-embedded picture data of M×M dots.

In **S012**, the CPU **1** writes the identification information watermark-embedded picture data generated in **S011** into the output disk unit **4**. Then, the CPU **1** terminates the watermark-embedding process.

[Structure of Extracting Computer]

Next, an explanation is given of a concrete structure of the extracting computer. FIG. **4** is an outline block diagram illustrating only a structure relative to a process for extracting identification information from the identification information watermark-embedded picture data in hardware of this extracting computer. As shown in FIG. **4**, the extracting computer is provided with a CPU (Central Processing Unit) **1**, an input disk unit **2**, a ROM (Read Only Memory) **3** and an output unit **5** which are connected one another by a bus B. That is, the hardware configuration of the extracting computer is similar to that of the watermark-embedding computer, however, there are some differences that an identification information extracting program **34** is stored in the ROM **3** and the output unit **5** is necessary but the output disk unit **4**. Thus, when both the identification information watermark-embedding program **31** and the identification information extracting program **34** are stored in the ROM **3**, and the output disk unit **4** and the output unit **5** are connected to the bus B. one computer can be used as the watermark-embedding computer and the extracting computer.

In FIG. **4**, the input disk unit **2** inputs process object picture data (identification information watermark-embedded picture data) **22** into the CPU **1**, such as a hard disk unit, a floppy disk unit and an optical magnetic disk unit.

The ROM **3** used as a watermark function hold unit and a computer readable medium is a read only memory holding an identification information extracting program **34** running in the CPU **1** and the identification information **32**. This identification information **32** is just the same in the watermark-embedding computer.

The CPU **1** is a processor controlling all of the extracting computer. The identification information extracting program **34** read from the ROM **3** runs, whereby an orthogonal transformation process part **16**, a coefficient extraction process part **17** and an identification information calculation part **18** are implemented in the CPU **1**, and the identification information extracting process shown in FIG. **5** is executed. Dot lines in FIG. **4** indicate data flows in the CPU **1**.

The orthogonal transformation process part **16** used as an orthogonal transformation unit applies the above-mentioned orthogonal transformation to the process object picture data **22** of N×M pixels, read from the input disk unit **2**, and then calculates a weight coefficient for each of N×M combinations of the basis functions. The orthogonal transformation process part **16** informs the coefficient extraction process part **17** of this coefficient distributed picture data including N×M weight coefficients.

The coefficient extraction process part **17** as a pickup unit extracts L-piece weight coefficients in which there is a possibility that signals of the identification information are watermark-embedded, and then informs the identification information calculation part **18** of them.

The identification information calculation part **18** used as a calculation unit reads a watermark-embedding function **33** from the ROM **3**. The output value of the watermark-embedding function **33** is obtained every weight coefficient informed from the coefficient extraction process part **17**. Then, the identification information calculation part **18** arranges each obtained output value in accordance with the arrangement in the coefficient distributed picture data of the weight coefficient corresponding to that output value, and then outputs them to the output unit **5**.

The output unit **5** is a display unit for displaying L-piece output values received from the CPU **1**, a printer for printing output values or the like.

FIG. **5** is a flow chart illustrating an identification information extracting process executed by the CPU **1** which has read the identification information extracting program **34**. The identification information extracting process starts by inputting an identification information extracting command via a keyboard (not shown) connected to the watermark-embedding computer.

In **S101** executed initially after starting the identification information extracting process, the CPU **1** reads L-piece watermark-embedding functions **33**($f_i(x)$ [i=1–L]) from the ROM **3**.

In **S102**, the CPU **1** reads the process object picture data of N×M pixels from the input disk unit **2**.

In **S103**, the CPU **1** applies the above mentioned orthogonal transformation process to all of the process object picture data read in **S102** so as to generate coefficient distributed picture data of N×M dots.

In **S104**, the CPU **1** selects a weight coefficient C_i [i=1–L] at a position corresponding to each signal S_i [i=1–L] of the identification information **32** in the watermark-embedding computer among weight coefficients included in the coefficient distributed picture data generated in **S103**.

In **S105**, the CPU **1** initializes a variable i to specify the weight coefficient C_i of a process object into "1".

Next, the CPU **1** executes a loop process between **S106** and **S108**. In **S106** to be the first step of this loop, the CPU **1** obtains an output value S_i for the weight coefficient C_i of the watermark-embedding function $f_i(x)$ corresponding to the weight coefficient C of the process object.

In **S107**, the CPU **1** checks whether the variable i gets to L or not, namely, whether the process in **S106** is applied to all weight coefficients to which identification information may be watermark-embedded or not. When the variable i does not yet get to L, the CPU **1** increases the variable i in **S108**, and then returns the process to the **S106**.

On the contrary, when the variable i gets to L, in **S109**, the CPU **1** arranges all output values S_i [i=1–L] obtained in **S106** in accordance with the arrangement in the coefficient distributed picture data of the corresponding weight coefficients C_i [i=1–L] so as to output them to the output unit **5**.

As the results, the output unit **5** can display or print data corresponding to the identification information when the process object picture data read in **S102** is identification information watermark-embedded picture data.

Additionally, two-dimensional DCT (Discrete Cosine Transformation), two-dimensional DST (Discrete Sine Transformation) or two-dimensional Hadamard's Transformation maybe used as the orthogonal transformation in this aspect. Next, an explanation will be given of the concrete processes in the identification information watermark-embedding process and the identification information extracting process when two-dimensional DCT is used as the orthogonal transformation with reference to Embodiment 1.

Embodiment

In this Embodiment, an original picture data includes N×N pixels (i.e., square of pikets; only, N>>8), as shown in FIG. **10**(*a*). The luminance value of each pixel in the original picture data is shown by the 0–255 gray scale. The identification information, as shown in FIG. **10**(*d*), is picture data (hereinafter, called "signature picture data") showing characters "FJ" by giving a white luminance value (255) or a black luminance value (0) selectively to each of 8×8 pixels. In addition to this, the identification information watermark-embedded picture data is called "signature picture watermark-embedded picture data". Since the weight coefficient obtained by the two-dimensional DCT corresponds to the intensity (amplitude) of each frequency component in the original picture data, the above-mentioned "coefficient distributed picture data" is called "frequency distributed picture data". As the watermark-embedding function, one function f(x) applied to respective weight coefficients in common, namely, a continuous periodic function in a saw tooth shown in a graph in FIG. **10**(*e*) is used.

(Identification Information Watermark-embedding Process)

FIGS. **6** and **8** are flow charts illustrating the identification information watermark-embedding process in the Embodiment.

In **S201** executed initially after staring the identification information watermark-embedding process, the CPU **1** reads signature picture data S (i,j) [i=0–7, j=0–7]) from the ROM **3**.

In **S202**, the CPU **1** reads the watermark-embedding function f(x) from the ROM **3**.

In **S203**, the CPU **1** reads the process object picture data of N×N pixels from the input disk unit **2**.

In **S204**, the CPU **1** applies the two-dimensional DCT to all of the original picture data read in **S203**, and then generates frequency distributed picture data of N×N dots shown in FIG. **10**(*b*). Concretely, in **S204**, the CPU **1** performs a two-dimensional DCT process subroutine shown in FIG. **7**.

In **S301** to be the first step of this two-dimensional DCT process subroutine, the CPU **1** initializes a variable i showing a column (0 is leftmost) in the frequency distributed picture data of the calculation object weight coefficient C(i,j) to be "0".

In **S302**, the CPU **1** sets a function $W_i(x)$ of a variable x as indicated by the following expression (2). In this expression, the variable x corresponds to a column (0 is leftmost) of each pixel in the original picture data.

$$W_o(x)=\sqrt{\frac{1}{N}} \tag{2}$$

In S303, the CPU 1 initialize a variable j showing a row (0 is highest) in the frequency distributed picture data of the calculation object weight coefficient C(i,j) to be "0".

In S304, the CPU 1 sets a function $W_i(y)$ of a variable y as indicated by the following expression (3). In this expression, the variable y corresponds to a row (0 is highest) of each pixel in the original picture data).

$$W_o(y)=\sqrt{\frac{1}{N}} \tag{3}$$

In S305, the CPU 1 executes the following expression (4) based on both functions $W_i(x)$ and $W_i(y)$ set at the current time and a luminance value G(x,y) of each pixel in the original picture data, and sets the calculated series as the weight coefficient C(i,j) specified by the current variables i and j.

$$C(i,j)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} G(x,y)\cdot W_i(x)\cdot W_j(y) \tag{4}$$

In S306, the CPU 1 checks whether the current variable j gets to (N−1) or not. When the variable j does not yet get to (N−1), the CPU 1 increases the variable j in S307. Then, the CPU 1 substitutes the current variable j to the following expression (5), and sets a new function $W_i(y)$ of a variable y again.

$$W_j(y)=\sqrt{\frac{2}{N}}\cos\frac{\pi(2y+1)}{2N}j \tag{5}$$

Thereafter, the CPU 1 returns the process to S305, and calculates the weight coefficient C(i,j) in the next row.

On the contrary, when it is determined that the current variable j gets to (N−1) in S306, the CPU 1 advances the process to S309. In S309, the CPU 1 checks whether the current variable i gets to (N−1) or not. When the variable i does not yet get to (N−1), the CPU 1 increases the variable i in S310. Then, the CPU 1 substitutes the current variable i to the following expression (6), and sets a new function $W_i(x)$ of a variable x again.

$$W_i(x)=\sqrt{\frac{2}{N}}\cos\frac{\pi(2x+1)}{2N}i \tag{6}$$

Thereafter, the CPU 1 returns the process to S303, and calculates the weight coefficient C(i,j) in the next column.

On the contrary, when it is determined that the current variable i gets to (N−1) in S309, the CPU 1 determines that all weight coefficients in the frequency distributed picture data are calculated, and then terminates this subroutine so as to return the process to the main routine in FIG. 6.

In the main routine in FIG. 6, the process is advanced to S205 after finishing S204. The processes between S205 and S212, as shown in FIGS. 10(*b*) through 10(*g*), are executed to pick up areas indicating DC components and areas indicating low frequency components (areas i=0–7 and j=0–7) in the frequency distributed picture data (FIG. 10(*b*)), and to watermark the signature picture data into those areas. The signature picture data is watermark-embedded into only weight coefficients indicating DC components and weight coefficients indicating low frequency components, because some variations of the DC components and low frequency components influence slightly on the picture quality of the signature picture watermark-embedded picture data.

In S205, the CPU 1 initializes the variable i indicating a column of the pickup object weight coefficient C(i,j) in the frequency distributed picture data and indicating a column of the reference object pixel S(i,j) in the signature picture so as to be "0".

In S206, the CPU 1 initializes the variable j indicating a row of the pickup object weight coefficient C(i,j) in the frequency distributed picture data and indicating a row of the reference object pixel S(i,j) in the signature picture so as to be "0".

In S207, the CPU 1 reads the signature image data (FIG. 10(*d*)) and the watermark-embedding function f(x) (FIG. 10(*e*)) from the ROM 3, and then obtains all input values c_{ijk} of the watermark-embedding function f(x) which takes a luminance value of the reference object pixel S(i,j) in the signature picture data specified by the current variables i and j for an output value. That is, all of K_{ij}-piece solutions c_{ijk} ($1\leq k\leq K_{ij}$) which satisfy $f(c_{ijk})=S(i,j)$ are obtained. Here, it is defined that the number of the solutions c_{ijk} for $f(c_{ijk})=S(i,j)$ is K_{ij}.

In S208, the CPU 1 selects an input value which is closest to the pickup object weight coefficient C(i,j) specified by the current variables i and j among all input values c_{ijk} obtained in S207. Then, the CPU 1 permutates the pickup object weight coefficient C(i,j) specified by the current variables i and j for the selected input value (see FIG. 10(*f*)).

In S209, the CPU 1 checks whether the current variable j gets to "7" or not. When the variable j does not yet get to "7", the CPU 1 increases the variable j in S210, and then returns the process to S207 to permutate the weight coefficient C(i,j) in-the next row.

On the contrary, when it is determined that the current variable j gets to "7" in S209, the CPU 1 advances the process to S211. In S211, the CPU 1 checks whether the current variable i gets to "7" or not. When the variable i does not yet get to "7", the CPU 1 increases the variable i in S212 and then returns the process to S206 to permutate the weight coefficient C(i,j) of the next column.

On the contrary, when it is determined that the current variable i gets to "7" in S211, the CPU 1 advances the process to S213. In S213, the CPU 1 applies the two-dimensional inverse DCT to all frequency distributed picture data (FIG. 10(*g*)) including the weight coefficient C(i,j) of which the value is permutated in S208 so as to generate the signature picture watermark-embedded picture data of N×N dots shown in FIG. 10(*h*). Concretely, in S213, the CPU 1 performs the two-dimensional inverse DCT process subroutine shown in FIG. 8.

In S401 to be the first step of this two-dimensional inverse DCT process subroutine, the CPU 1 defines functions used for this process like the following expressions (7)–(10).

$$W_o(x)=\sqrt{\frac{1}{N}} \tag{7}$$

$$W_i(x)=\sqrt{\frac{2}{N}}\cos\frac{\pi(2x+1)}{2N}i \tag{8}$$

-continued

$$W_o(y) = \sqrt{\frac{1}{N}} \tag{9}$$

$$W_j(y) = \sqrt{\frac{2}{N}} \cos\frac{\pi(2y+1)}{2N} j \tag{10}$$

In S402, the CPU 1 initializes the variable x indicating a column (0 is leftmost) in the signature picture watermark-embedded picture data of the calculation object pixel R(x,y) so as to be "0".

In S403, the CPU 1 initializes the variable y indicating a row (0 is highest) in the signature picture watermark-embedded picture data of the calculation object pixel R(x,y) so as to be "0".

In S404, the CPU 1 calculates the luminance value of the pixel R(x,y) in the signature picture watermark-embedded picture data specified by the current variables x and y. Concretely, the CPU 1 substitutes the current variable x to the function of the expression (8) defined in S401, and substitutes the current variable y to the function of the expression (10) defined in S401. Moreover, the CPU 1 executes the following expression (11) based on the functions of the substituted expressions (8) and (10) and the functions of the expressions (7) and (9) defined in S401, and then sets the calculated series as the luminance value of the pixel R (x,y) specified by the current variables x and y.

$$R(x, y) = \sum_{i=0}^{N-1} \sum_{i=0}^{N-1} C(i, j) \cdot W_i(x) \cdot W_j(y) \tag{11}$$

In S405, the CPU 1 checks whether the current variable y gets to (N−1) or not. When the variable y does not get yet to (N−1), the CPU 1 increases the variable y in S406, and then returns the process to S404 so as to calculate the luminance value of the pixel R (x,y) in the next row.

On the contrary, when it is determined that the current variable y gets to (N−1) in S405, the CPU 1 advances the process to S407. In S407, the CPU 1 checks whether the current variable x gets to (N−1) or not. When the variable x does not yet get to (N−1), the CPU 1 increases the variable x in S408, and then returns the process to S403 so as to calculate the luminance value of pixel R (x,y) in the next column.

On the contrary, when it is determined that the current variable x gets to (N−1) in S407, the CPU 1 determines that all luminance value of pixels in the signature picture watermark-embedded picture data are calculated, and then terminates this subroutine to return the process to the main routine in FIG. 6.

In the main routine in FIG. 6, the process is advanced to S214 after finishing S213. In S214, the CPU 1 outputs the signature picture watermark-embedded picture data of N×N pixels to the output disk unit 4.

(Identification Information Extracting Process)

FIG. 9 is a flow chart illustrating an identification information extracting process in the Embodiment.

In S501 to be the first step of this identification information extracting process, the CPU 1 reads the watermark-embedding function f(x) from the ROM 3.

In S502, the CPU 1 reads the process object picture data of N×N pixels shown in FIG. 11(*a*) from the input disk unit 2.

In S503, the CPU 1 applies the two-dimensional DCT to all process object picture data read in S502 so as to generate the frequency distributed picture data of N×N pixels shown in FIG. 11(*b*). Concretely, the CPU 1 calculates the two-dimensional DCT subroutine shown in FIG. 7.

In the processes S504 through S510, as shown in FIGS. 11(*b*) through 11(*e*), areas indicating DC components and areas indicating low frequency components (area of i=0–7 and j=0–7) (FIG. 11(*c*)) in the frequency distributed picture data (FIG. 11(*b*)) are picked up, and the signature picture data is extracted from those areas.

In S504, the CPU 1 initializes the variable i indicating a column of the pickup object weight coefficient C (i,j) in the frequency distributed picture data so as to be "0".

In S505, the CPU 1 initialize the variable j indicating a row of the pickup object weight coefficient C (i,j) in the frequency distributed picture data so as to be "0".

In S506, the CPU 1 reads a watermark-embedding function (FIG. 11(*d*) equal to FIG. 10(*e*)), and then substitutes the extraction object weight coefficient C (i,j) specified by the current variables i and j i.e., evaluates the watermark-embedding function at the value C(i,j) so as to calculate an output value S'(i,j) of the ith column and the jth row.

In S507, the CPU 1 checks whether the current variable j gets to "7" or not. When the variable j does not yet get to "7", the CPU 1 increases the variable j in S208, and then returns the process to S506 to calculate the output value S' in the next column.

On the contrary, when it is determined that the current variable j gets to "7" in S507, the CPU 1 advances the process to S509. In S509, the CPU 1 checks whether the current variable i gets to "7" or not. When the variable i does not yet get to "7", the CPU 1 increases the variable i in S510, and then returns the process to S505 to calculate the output value S'(i,j) in the next column.

On the contrary, when it is determined that the current variable i gets to "7" in S509, the CPU 1 advances the process to S511. In S511, the CPU 1 outputs output values (luminance values) of i×j dots calculated in S506 to the output unit 5. At that time, when the process object picture data is the signature picture watermark-embedded picture data, that picture data is the same picture data (FIG. 11(*e*)) as the signature picture (FIG. 10(*d*)).

According to the present invention, it is possible to watermark identification information into an original picture data without deteriorating the picture quality of the picture data so as not to be recognized by a third person. Further, it is also possible to extract the watermark-embedded identification information without the original picture data. Thus, since it is unnecessary to keep and manage original picture data for a right holder or the like of picture data, no mass storage unit is needed.

This invention being thus described, it will be obvious that same may be varied in various ways. Such variations are not to be regarded as departing from the spirit and scope of the invention, and all such modifications as would be obvious for one skilled in the art are intended to be included within the scope of the following claims.

What is claimed is:

1. A method of watermark-embedding identification information into original picture data comprising a plurality of pixel values, the identification information comprising numerical signals representing respective identification values, the method comprising:

generating combinations of mutually orthogonal basis functions for the identification values;

calculating corresponding weight coefficients for each of the combinations, each weight coefficient further corresponding to a basis function of the corresponding

combination and being calculated based on a sum of products respectively corresponding to the pixels, each product being calculated by multiplying the pixel value of the pixel and a value of the basis function at a position of the pixel within the original picture data;

specifying for each identification value corresponding input values for a watermark-embedding function defined as a multi-to-one function having a domain comprising the weight coefficients and a range comprising the identification values, each specified input value being closest to a corresponding weight coefficient of the identification value among plural input values mapped to the identification value by the multi-to-one function; and

changing the weight coefficients to values equal to the corresponding specified input values by changing pixel values in the original picture data.

2. A method of extracting identification information from process object picture data into which identification information is watermark-embedded by the watermark-embedding method according to the claim 1, said extracting method comprising:

generating combinations of mutually orthogonal basis functions respectively corresponding to the identification values;

calculating corresponding weight coefficients for each of the respective combinations, each calculated weight coefficient further corresponding to a basis function of the combination and being based on a sum of products respectively corresponding to pixels of the process object picture data, each product being calculated by multiplying a pixel value of the pixel and a value of the basis function at a corresponding pixel position within the original picture data; and

calculating for each weight coefficient a value of a watermark-embedding function defined as a multi-to-one function having a domain comprising the weight coefficients and a range comprising the identification values.

3. An extracting method according to claim 2, wherein the watermark-embedding function is a periodic function.

4. An extracting method according to claim 2, wherein the watermark-embedding function is a continuous periodic function.

5. An identification information watermark embedding method according to claim 2, wherein the watermark-embedding function transforms an interval between two input values into an image interval when the watermark-embedding function maps the two input values to a same output value, the image interval being narrow when the two input values are small and being broad when the two input values are large.

6. A watermark-embedding method according to claim 1, wherein the watermark-embedding function is a periodic function.

7. A watermark-embedding method according to claim 1, wherein the watermark-embedding function is a continuous periodic function.

8. A watermark-embedding method according to claim 1, wherein the watermark-embedding function transforms an interval between two input values into an image interval when the watermark embedding function maps the two input values to a same output value, the image interval being narrow when the two input values are small and being broad when the two input values are large.

9. A method of watermark-embedding identification information into original picture data comprising a plurality of pixel values, the identification information comprising numerical signals representing identification values, the method comprising:

applying orthogonal transformation to the plurality of pixel values, and generating a plurality of weight coefficients;

designating weight coefficients for each of the identification values, the designated weight coefficients being selected from the plurality of weight coefficients;

specifying for the weight coefficients of each identification value corresponding input values for a watermark-embedding function defined as a multi-to-one function having a domain comprising the weight coefficients and a range comprising the identification values, each specified input value being closest to the corresponding weight coefficient among plural input values mapped to the identification value by the multi-to-one function, and permutating the designated weight coefficients by the corresponding specified input values; and

applying inverse orthogonal transformation to the plurality of weight coefficients after permutating the weight coefficients corresponding to all the identification values.

10. A method of extracting identification information from process object picture data to which the identification information is watermark-embedded by the watermark-embedding method according to claim 9, the extracting method comprising:

applying orthogonal transformation to pixel values of the process object picture data and generating of a plurality of weight coefficients;

selecting weight coefficients corresponding to the identification values from the plurality of weight coefficients; and

calculating for each selected weight coefficient a value of a corresponding watermark-embedding function defined as a multi-to-one function having a domain comprising the weight coefficients and having a range comprising the identification values.

11. A method of watermark-embedding identification information into original picture data comprising a plurality of pixel values, the identification information comprising numerical signals representing identification values, the method comprising:

applying two-dimensional discrete cosine transformation to the plurality of pixel values, and generating a plurality of weight coefficients;

designating weight coefficients for each of the identification values, the designated weight coefficients being selected from the plurality of weight coefficients;

specifying for the designated weight coefficients of each identification value corresponding input values for a watermark-embedding function defined as a multi-to-one function having a domain comprising the weight coefficients and a range comprising the identification values, each specified input value being closest to the corresponding weight coefficient among plural input values mapped to the identification value by the watermark-embedding function, and permutating the weight coefficients; and

applying inverse two-dimensional discrete cosine transformation to the plurality of weight coefficients after permutating the weight coefficients corresponding to all the identification values.

12. A method of extracting identification information from process object picture data into which identification

information is watermark-embedded by the watermark-embedding method according to claim 11 the extracting method comprising:

applying two-dimensional discrete cosine transformation to pixel values of the process object picture data, and generating a plurality of weight coefficients;

selecting weight coefficients corresponding to the identification values from the plurality of weight coefficients; and

calculating for each selected weight coefficient a value of a watermark-embedding function defined as a multi-to-one function having a domain comprising the weight coefficients and having a range comprising the identification values.

13. A watermark-embedding apparatus for watermark embedding identification information into original picture data comprising a plurality of pixel values, the identification information comprising numerical signals representing identification values, said watermark-embedding apparatus comprising:

watermark-embedding function hold means for holding a watermark-embedding function defined as a multi-to-one function having a domain comprising a set of weight coefficients and having a range comprising the identification values;

orthogonal transformation means for applying orthogonal transformation to the plurality of pixel values, and generating a plurality of weight coefficients comprised in the set of weight coefficients;

weight coefficient permutation means for designating weight coefficients from the plurality of weight coefficients for each identification value, specifying for each designated weight coefficient a corresponding input value closest to the weight coefficient among plural input values mapped to the identification value by the watermark-embedding function, and permutating each designated weight coefficient by the corresponding specified input value; and

inverse orthogonal transformation means for applying inverse orthogonal transformation to the plurality of weight coefficients after permutation of the designated weight coefficients.

14. An extracting apparatus for extracting identification information from process object picture data in which the identification information is watermark-embedded by the watermark-embedding apparatus according to claim 13, said extracting apparatus comprising:

watermark-embedding function hold means for holding a watermark-embedding function defined as a multi-to-one function having a domain comprising a set of weight coefficients and a range comprising the identification values;

orthogonal transformation means for applying orthogonal transformation to pixel values of the process object picture data and generating therefrom a plurality of weight coefficients;

means for selecting weight coefficients corresponding to the identification values from the plurality of weight coefficients; and

calculation means for calculating a value of the watermark-embedding function for each of the selected weight coefficients.

15. A computer readable medium storing a program to control a computer, said program comprising instructions for:

applying orthogonal transformation to a plurality of pixel values and generating a plurality of weight coefficients;

designating weight coefficients from the plurality of weight coefficients for each of plural identification values represented by respective numerical signals;

specifying for the designated weight coefficients of each identification value corresponding input values for a watermark-embedding function defined as a multi-to-one function having a domain comprising the plurality of weight coefficients and having a range comprising the identification values, and permutating the designated weight coefficients in the plurality of weight coefficients by the corresponding specified input values, each specified input value being mapped to the identification value by the multi-valued function and being closest to the corresponding weight coefficient among plural input values mapped to the identification value by the watermark-embedding function; and

applying inverse orthogonal transformation to the plurality of weight coefficients after permutating the designated weight coefficients for all the identification values.

16. A computer readable medium storing a program to control a computer, said program comprising instructions for:

applying orthogonal transformation to a plurality of pixel values of process object picture data into which identification information is watermark-embedded, and generating a plurality of weight coefficients;

selecting from the plurality of weight coefficients corresponding weight coefficients for each of plural numerical signals representing respective specified values; and

for each selected weight coefficient, determining a corresponding domain value of a watermark-embedding function defined as a multi-to-one function having a domain comprising the plurality of weight coefficients, having a range comprising the specified values, and mapping each selected weight coefficient to the corresponding specified value; and

modifying the plurality of pixel values based on the determined domain values to output identification data.

17. A computer readable medium storing a program to control a computer, said program comprising instructions for:

applying orthogonal transformation to a plurality of pixel values of original picture data, and generating a plurality of weight coefficients;

selecting from the plurality of weight coefficients corresponding weight coefficients for each of plural numerical signals representing respective identification values; and

for each selected weight coefficient, determining a corresponding domain value of a watermark-embedding function defined as a multi-to-one function having a domain comprising the plurality of weight coefficients, having a range comprising the specified values, and mapping each selected weight coefficient to the corresponding identification value; and

modifying the plurality of pixel values based on the determined domain values to output watermark-embedded picture data.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO.: 6,104,826
DATED : August 15, 2000
INVENTOR(S): Akira NAKAGAWA, et al.

It is certified that errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Col. 20, line 30, after "generating" delete "of".

Col. 21, line 2, after "claim 11" insert --,--.

Signed and Sealed this

Twenty-fourth Day of April, 2001

Attest:

Nicholas P. Godici

NICHOLAS P. GODICI

Attesting Officer

Acting Director of the United States Patent and Trademark Office

US006285775B1

(12) **United States Patent**
Wu et al.

(10) **Patent No.: US 6,285,775 B1**
(45) **Date of Patent: Sep. 4, 2001**

(54) **WATERMARKING SCHEME FOR IMAGE AUTHENTICATION**

(75) Inventors: **Min Wu; Bede Liu**, both of Princeton, NJ (US)

(73) Assignee: **The Trustees of the University of Princeton**, Princeton, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/164,474**

(22) Filed: **Oct. 1, 1998**

(51) **Int. Cl.**[7] **G06K 9/00**

(52) **U.S. Cl.** **382/100**

(58) **Field of Search** 382/100, 232; 380/28, 29, 30, 54; 705/51

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,530,759	*	6/1996	Braudaway et al.	380/54
5,721,788		2/1998	Powell	382/100
5,778,102	*	7/1998	Standford, II et al.	382/251
5,875,249		2/1999	Mintzer et al.	380/54
5,930,369	*	7/1999	Cox et al.	380/54
5,960,081	*	9/1999	Vynne et al.	713/176
6,061,793	*	5/2000	Tewfik et al.	713/176

OTHER PUBLICATIONS

BYTE Magazine, "How to Embed a Watermark", Jan. 1997, 1 page (http://www.byte.com/art/970/sec18/art3.htm).

Ahumada, Jr., "Luminance-Model-Based DCT Quantization for Color Image Compression", *SPIE Human Vision, Visual Processing, and Digital Display III*, 1992, 1666, 365–374.

Friedman, G.L., "The trustworthy Digital Camera: Restoring Credibility to the Photographic Image", *IEEE Trans. on Consumer Electronics*, Nov. 1993, 39(4), 905–910.

Kesavan, H., "EE 392c Autumn 1997 Final Project: Choosing a DCT Quantization Matrix for JPEG Encoding", http://www-ise.stanford.edu/class/ee392c/demos/kesavan, Jul. 1998, 5 pages.

Koch, E. et al., "Towards Robust and Hidden Image Copyright Labeling", *IEEE Workshop on Nonlinear Signal and Image Processing*, 1995, 1–4.

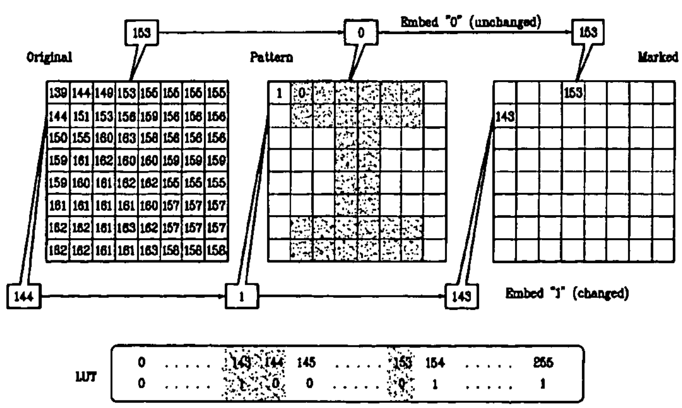
(List continued on next page.)

Primary Examiner—Andrew W. Johns
Assistant Examiner—Shervin Nakhjavan
(74) *Attorney, Agent, or Firm*—Woodcock Washburn Kurtz Mackiewicz & Norris LLP

(57) **ABSTRACT**

A digital watermarking process whereby an invisible watermark inserted into a host image is utilized to determine whether or not the image has been altered and, if so, where in the image such alteration occurred. The watermarking method includes the steps of providing a look-up table containing a plurality of coefficients and corresponding values; transforming the image into a plurality of blocks, wherein each block contains coefficients matching coefficients in the look-up table; and embedding the watermark in the image by performing the following substeps for at least some of the blocks: First, a coefficient is selected for insertion of a marking value representative of a corresponding portion of the watermark. Next, the value of the selected coefficient to used to identify a corresponding value in the look-up table. Finally, the identified coefficient is left unchanged if the corresponding value is the same as the marking value, and is changed if the corresponding value is different from the marking value. After the insertion of the watermark, the image may be stored in a lossy-compression form, thus permitting efficient storage and distribution. Moreover, the method may be used to produce two output signals for authentication: (1) a meaningful pattern to facilitate a quick visual check, and (2) an additional signal to detect unauthorized alteration. The method can be applied to an image compressed using JPEG or other techniques, such as Wavelet compression, and the marked image can be kept in the compressed format. Any alteration made on the marked image can be localized, making the method suitable for use in a "trustworthy" digital camera or camcorder.

14 Claims, 13 Drawing Sheets



OTHER PUBLICATIONS

Mintzer, F. et al., "Effective and Ineffective Digital Watermarks", *ICIP*, 1997, vol. 3, 4 pages.

Richter, J., "The Digital Watermark", http://www.richterscale.org/pcgr/pc960220.htm, May 1998, 3 pages.

Schneidner, M. et al., "A Robust Content Based Digital Signature for Image Authentication", *IEEE*, 1996, 227–230.

Storck, D., "A New Approach to Integrity of Digital Images", *IFIP Conf. on Mobile Communication*, 1996, 8 pages.

Swanson, M.D. et al., "Robust Data Hiding for Images", *IEEE Digital Signal Processing Workshop*, Sep. 1996, 37–40.

Wallace, G.K., "The JPEG Still Picture Compression Standard", *IEEE Trans. on Consumer Electronics*, 1991, 1–17.

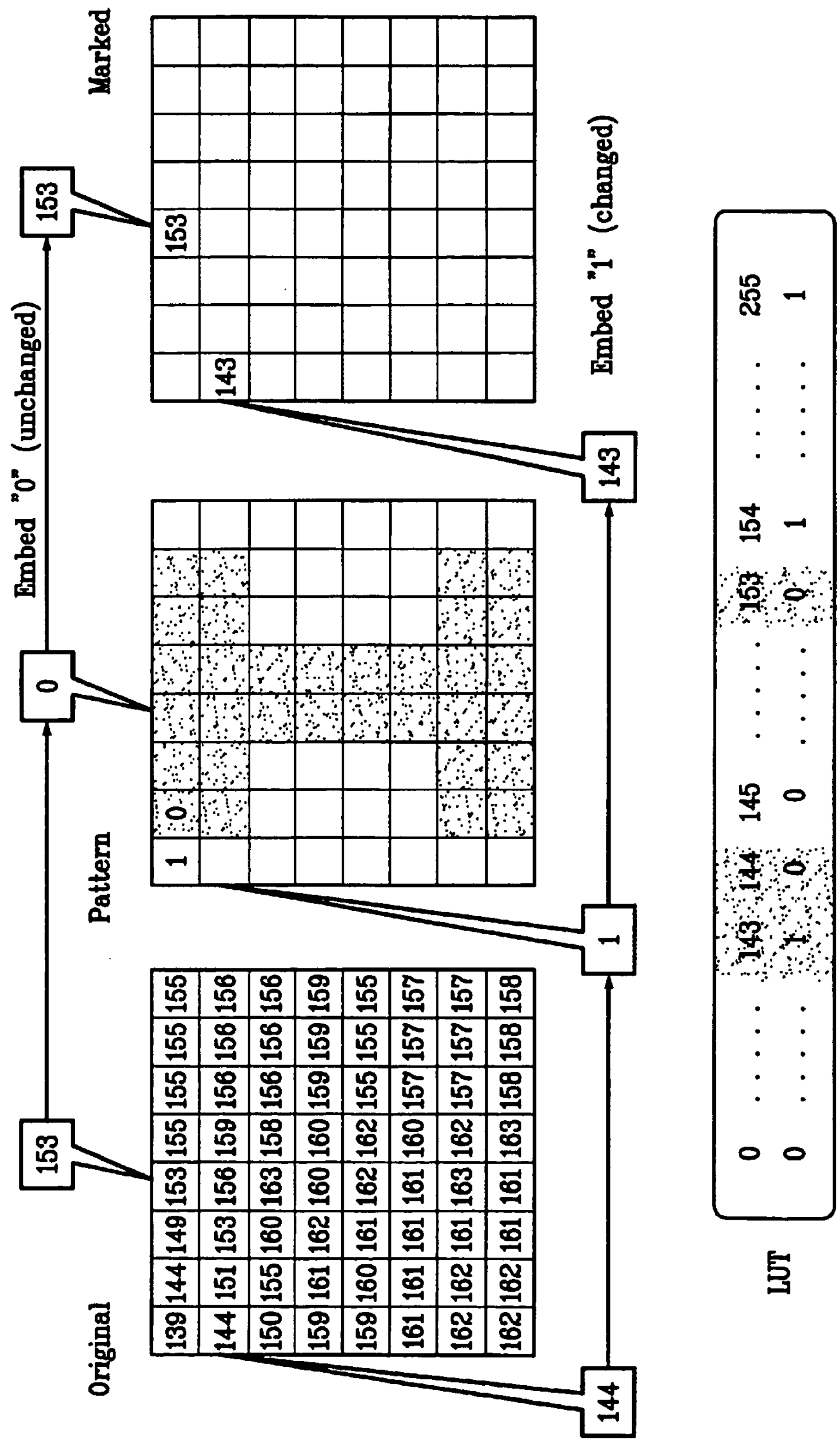
Yeung, M.M. et al., "An Invisible Watermarking Technique for Image Verification", *ICIP*, 1997, vol. 2, 4 pages,

Zeng, W, et al., "On Resolving Rightful Ownerships of Digital Images by Invisible Watermarks", *ICIP*, 1997, 4 pages.

Zhao, J., "Digital watermarking is the best way to protect intellectual property from illicit copying", (http://www.byte.com/art/9701/sec18/art1.htm), May 1998, 5 pages.

* cited by examiner

FIG. 1


Original
Pattern
Marked
Embed "0" (unchanged)
Embed "1" (changed)
LUT
153
0
153
144
1
143

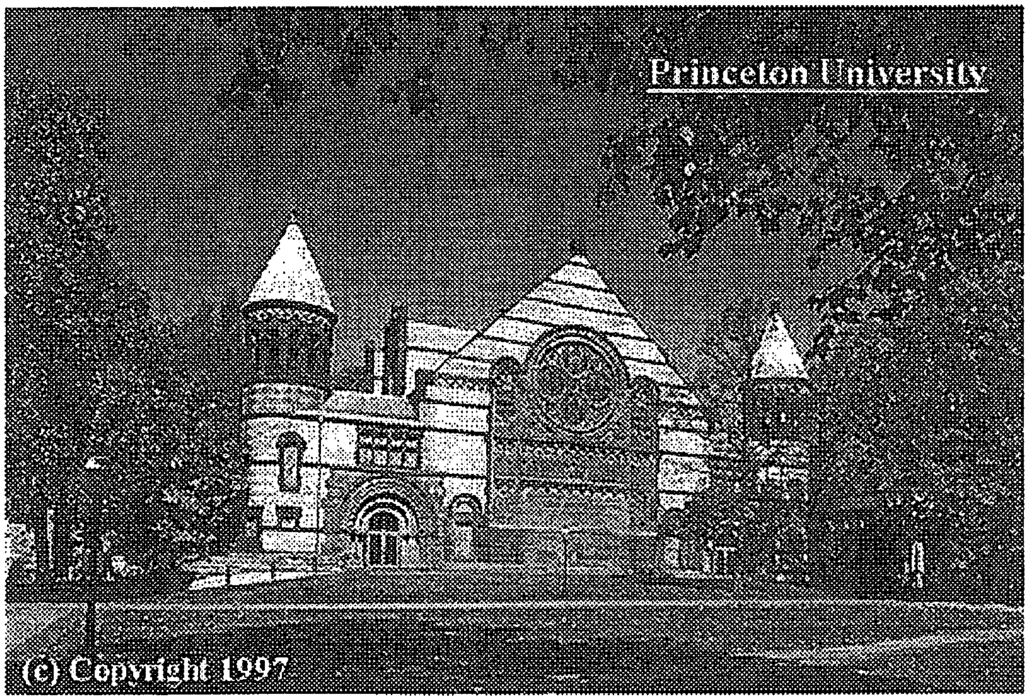


FIG. 2A

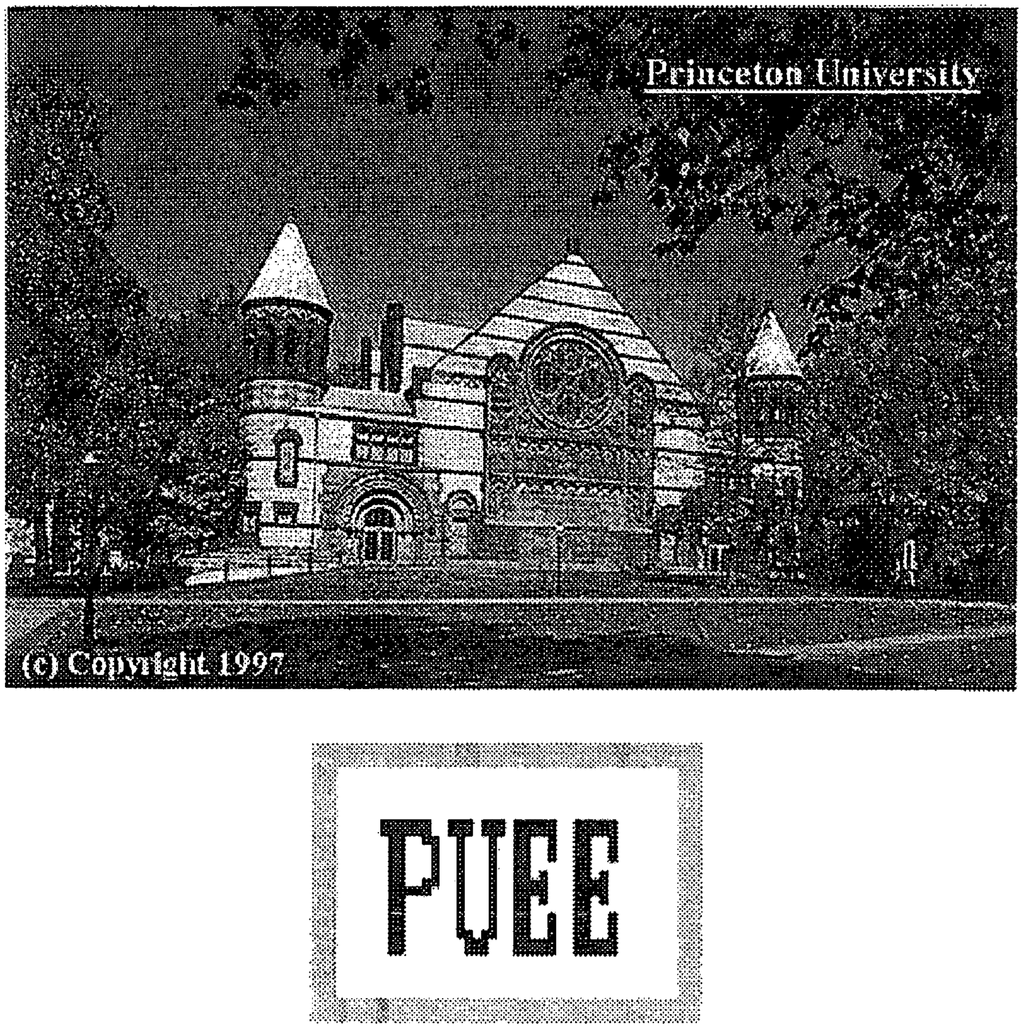


FIG. 2B

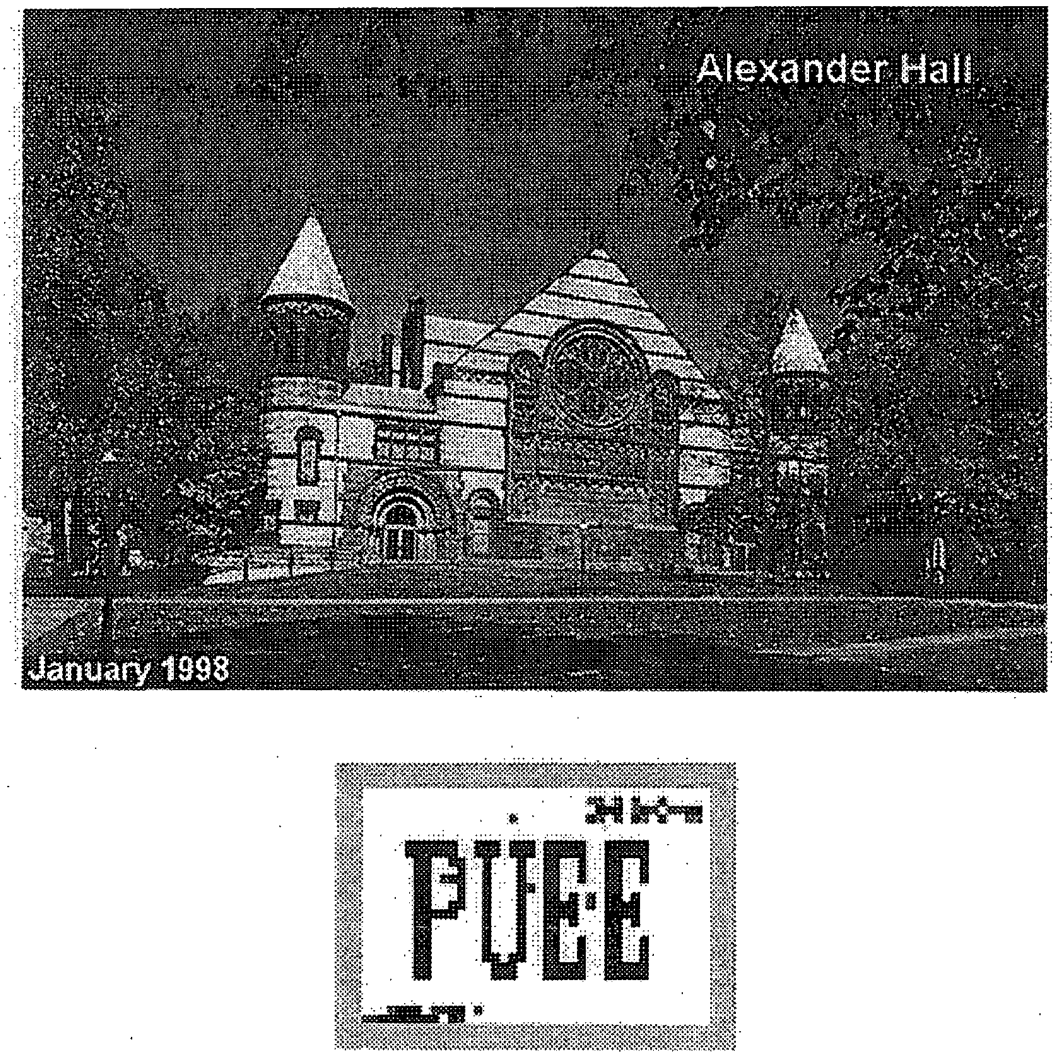


FIG. 2C

FIG. 3

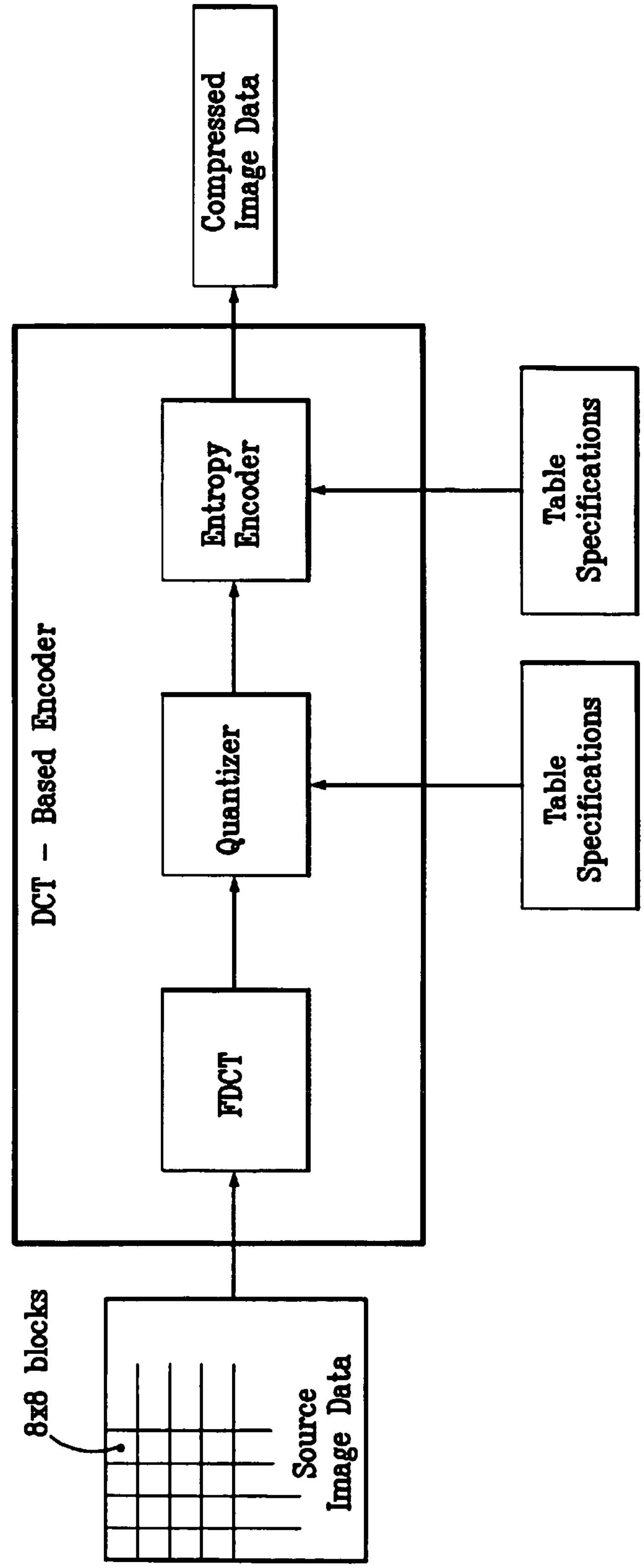

Compressed Image Data
DCT - Based Encoder
Entropy Encoder
Quantizer
FDCT
Table Specifications
Table Specifications
8x8 blocks
Source Image Data

FIG. 4

FIG. 5

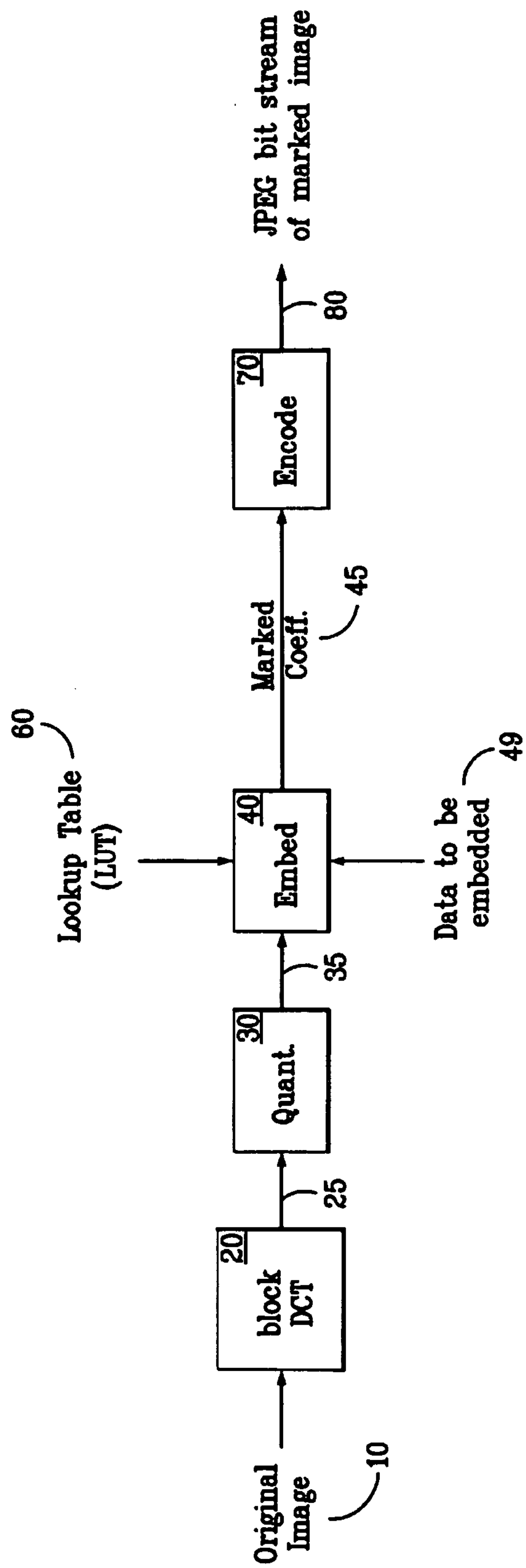
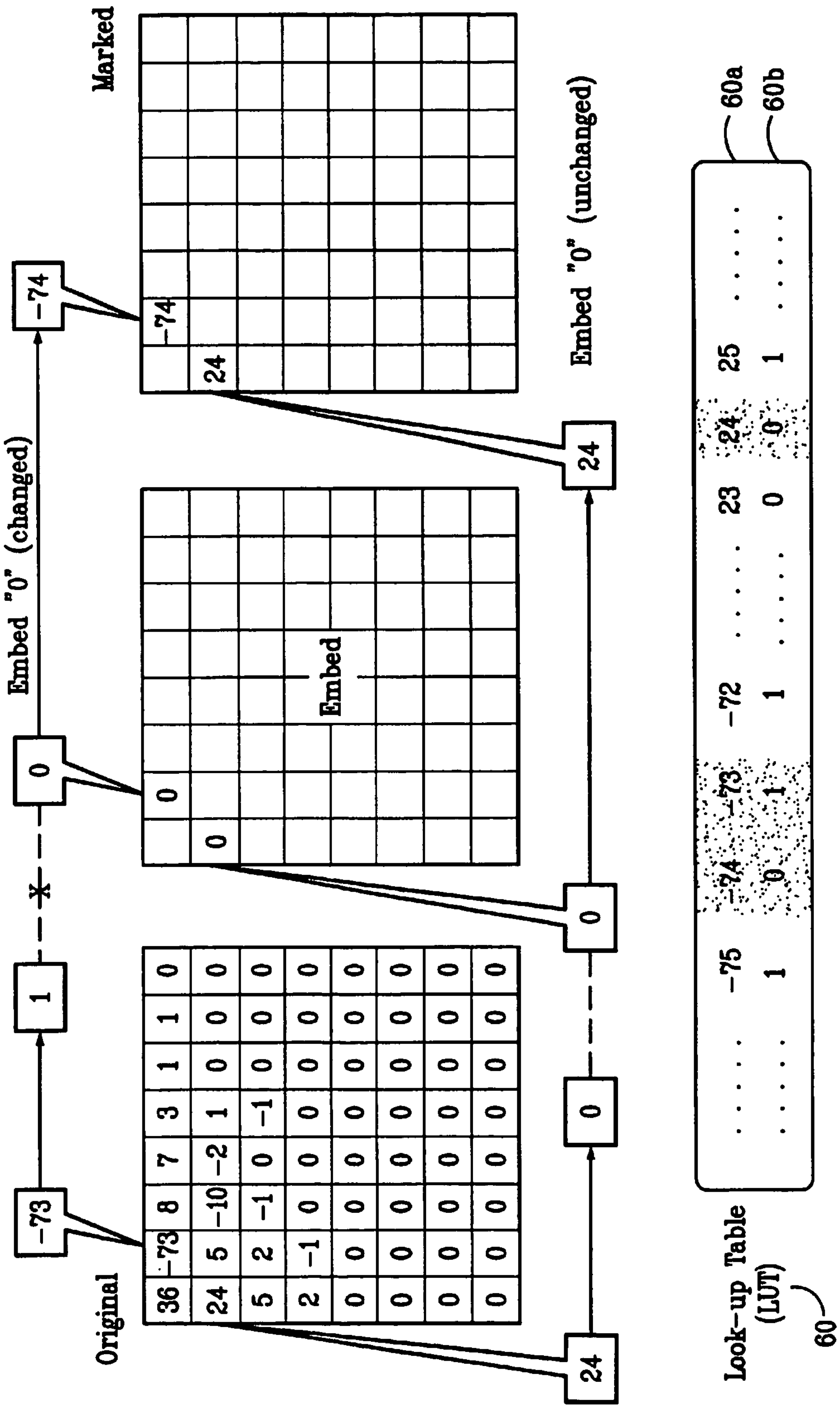

Original Image
10
block DCT
20
25
Quant.
30
35
Embed
40
Lookup Table (LUT)
60
Data to be embedded
49
Marked Coeff.
45
Encode
70
80
JPEG bit stream of marked image

FIG. 6


Original
Embed
Marked
Embed "0" (changed)
Embed "0" (unchanged)
Look-up Table (LUT)
60
60a
60b

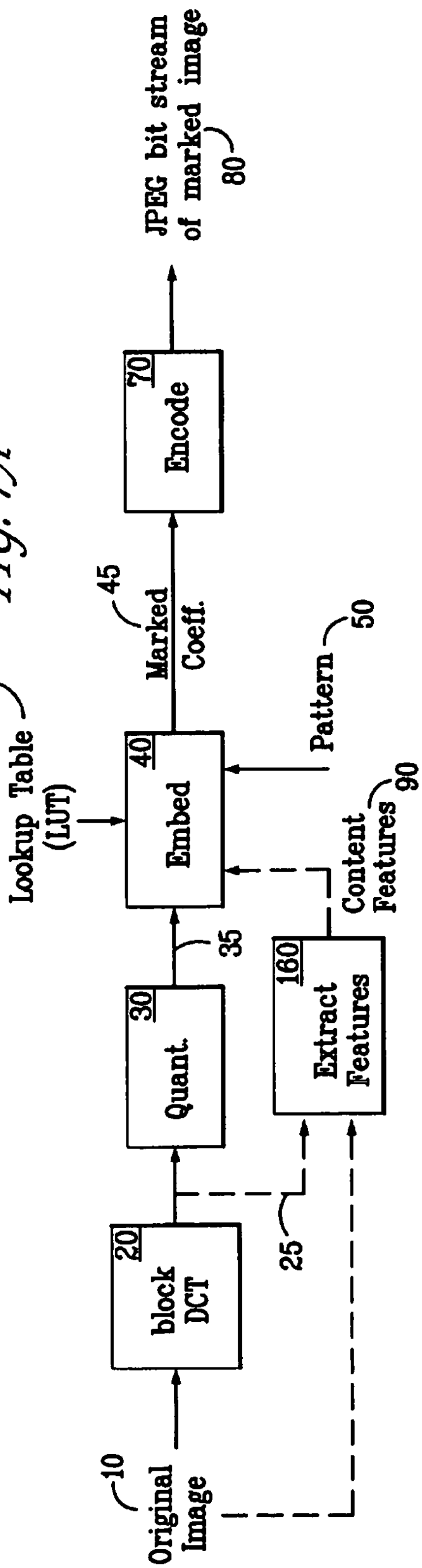

Original Image
10
block DCT
20
25
Quant.
30
35
Extract Features
160
Lookup Table (LUT)
60
Embed
40
Content Features
90
Pattern
50
Marked Coeff.
45
Encode
70
JPEG bit stream of marked image
80

FIG. 7A

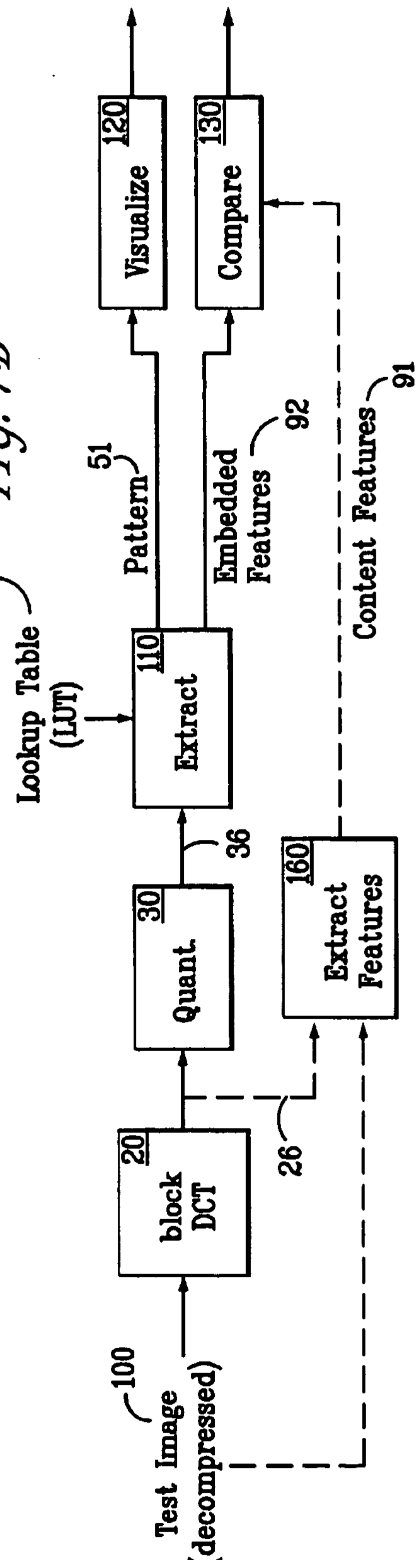

Test Image (decompressed)
100
block DCT
20
26
Quant.
30
36
Extract Features
160
Lookup Table (LUT)
60
Extract
110
Pattern
51
Embedded Features
92
Content Features
91
Visualize
120
Compare
130

FIG. 7B

FIG. 8A

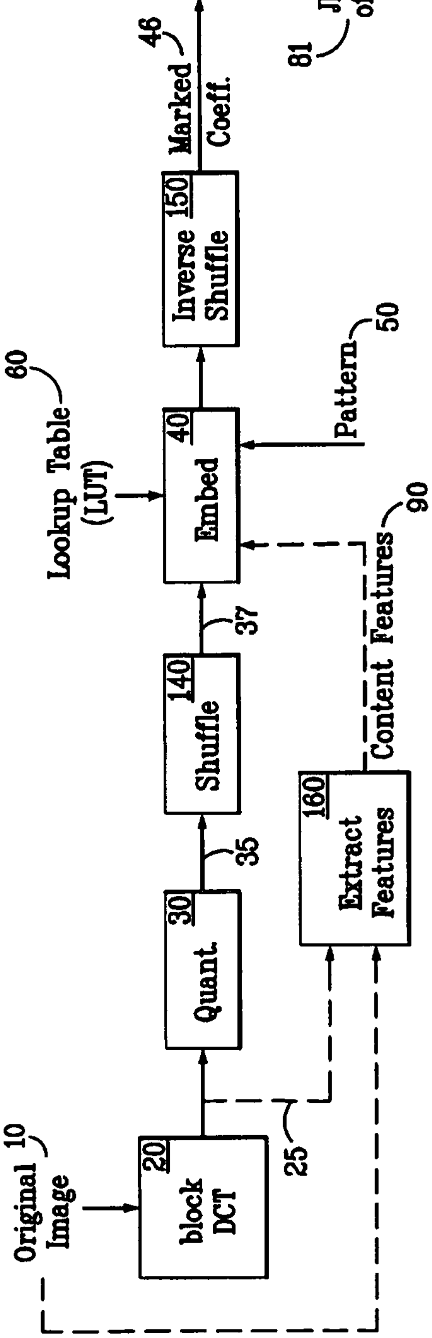

Original Image
10
20
block DCT
25
30
Quant.
35
140
Shuffle
37
160
Extract Features
Content Features
90
Lookup Table (LUT)
60
40
Embed
Pattern
50
150
Inverse Shuffle
Marked Coeff.
46
70
Encode
81
JPEG bit stream of marked image

FIG. 8B

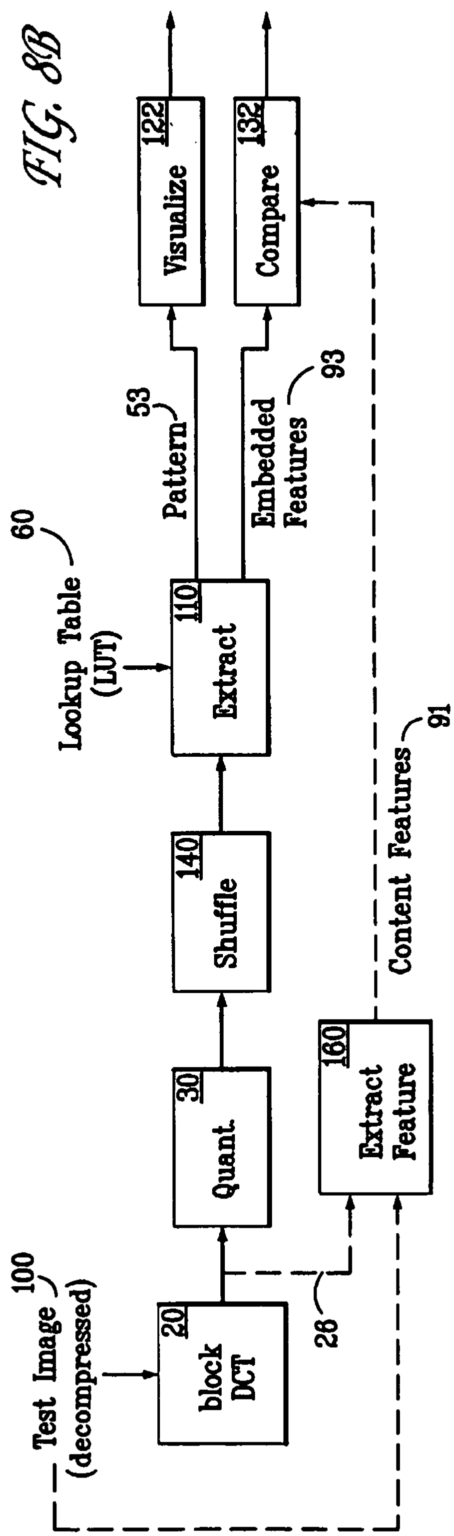

Test Image (decompressed)
100
20
block DCT
28
30
Quant.
140
Shuffle
160
Extract Feature
Content Features
91
Lookup Table (LUT)
60
110
Extract
Pattern
53
Embedded Features
93
122
Visualize
132
Compare

FIG. 9

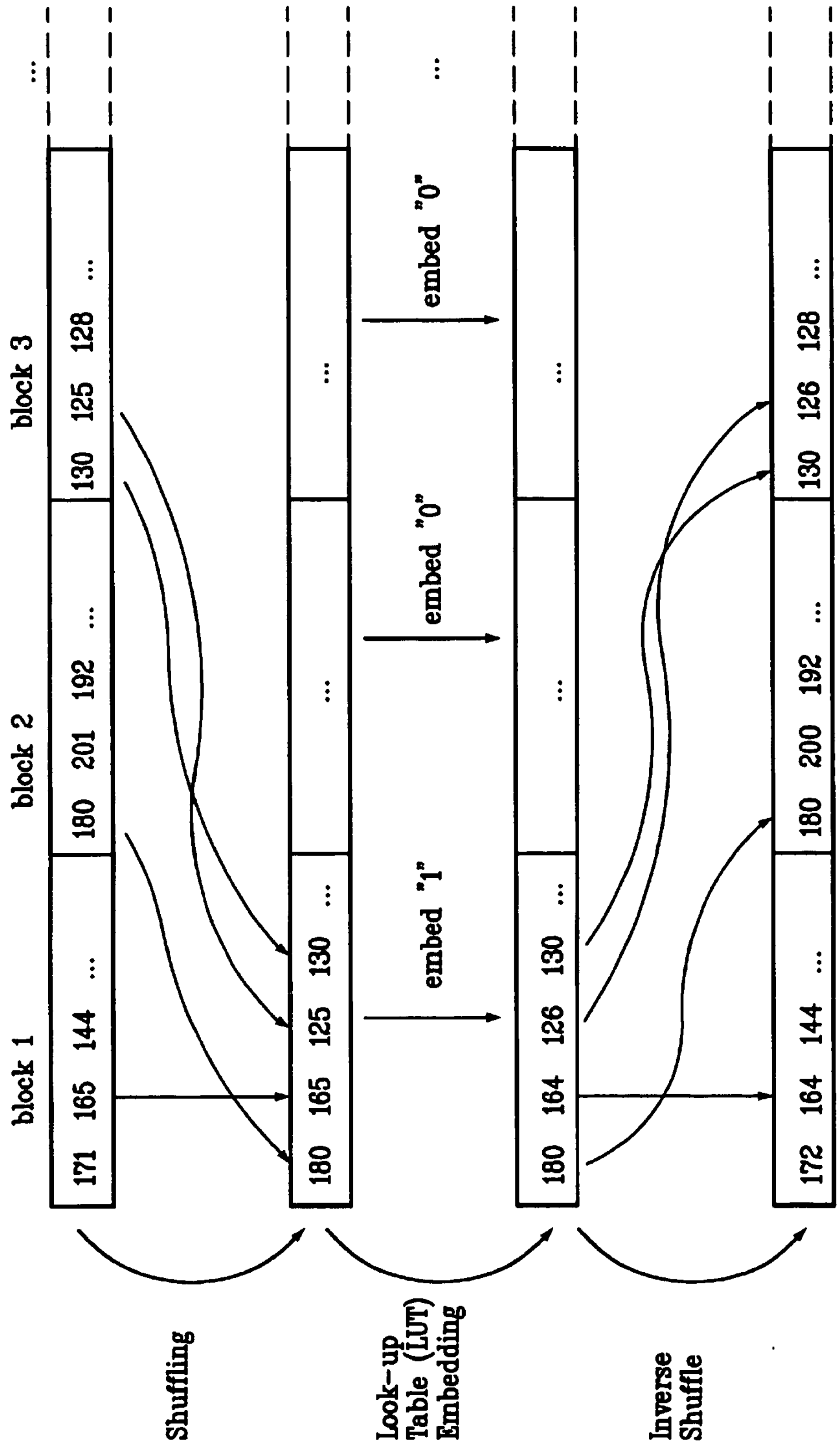

block 1
block 2
block 3
171 165 144 ...
180 201 192 ...
130 125 128 ...
180 165 125 130 ...
embed "1"
embed "0"
embed "0"
180 164 126 130 ...
172 164 144 ...
180 200 192 ...
130 126 128 ...
Shuffling
Look-up Table (LUT) Embedding
Inverse Shuffle

FIG. 10

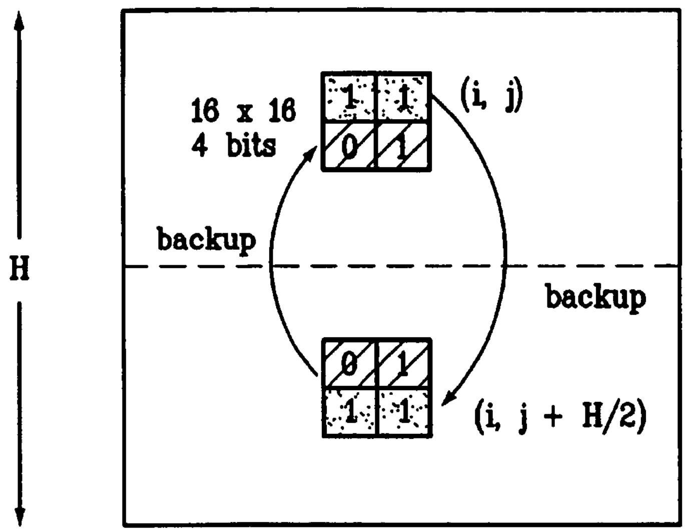
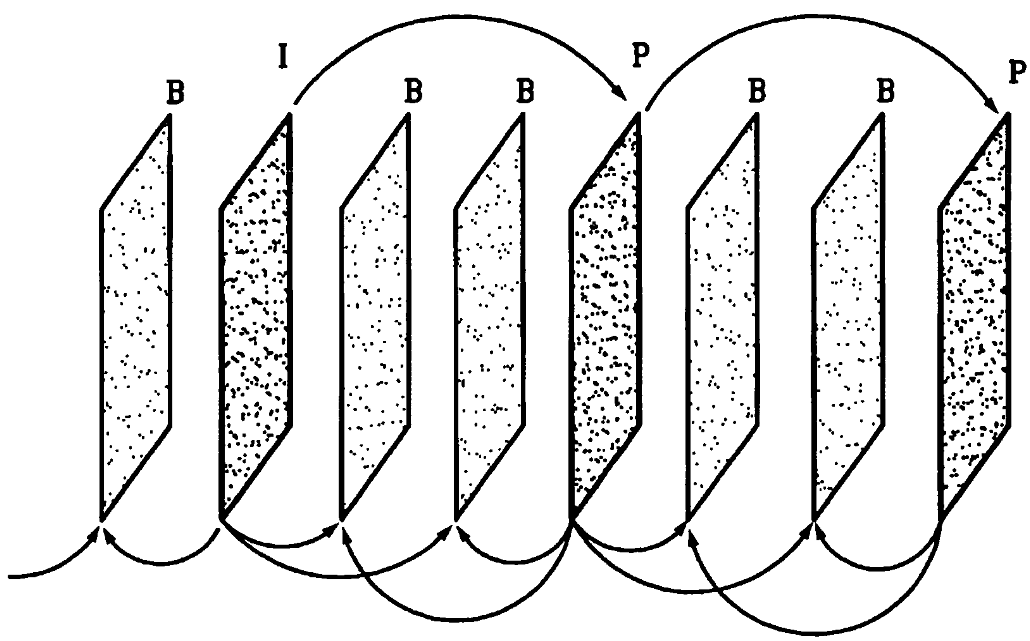


FIG. 11



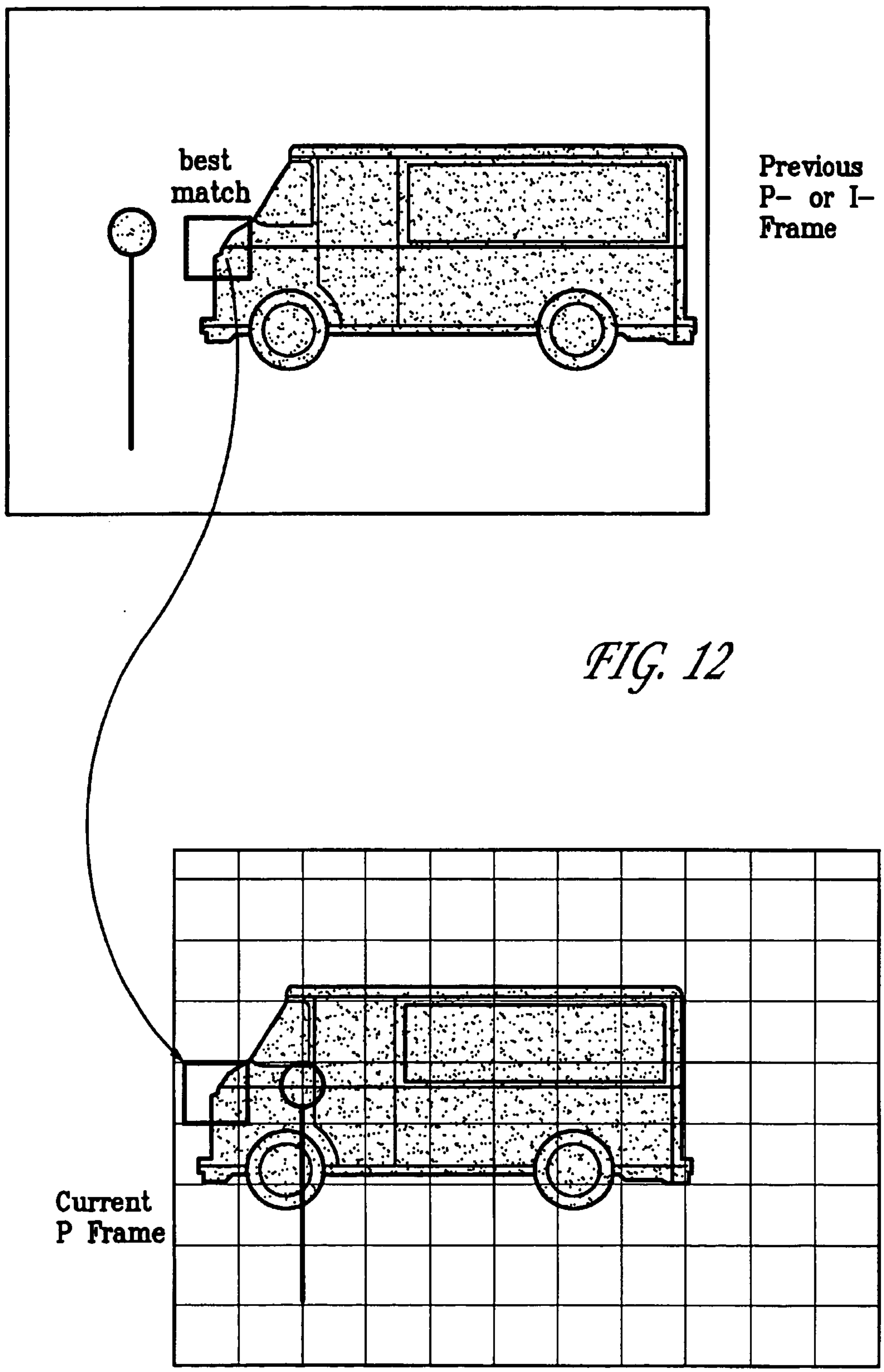


FIG. 12

WATERMARKING SCHEME FOR IMAGE AUTHENTICATION

I. FIELD OF THE INVENTION

The present invention concerns a new method and system for watermarking digital images for authentication purposes.

II. BACKGROUND OF THE INVENTION

A black-white or grey scale image of size X×Y can be described by its brightness in a plane, i.e., by a function f(x,y) where f is the brightness at a point with coordinates (x,y). A digital grey scale image comprises pixels arranged in a raster. Such an image can be described by f(i,j) where the brightness f takes on a value from a discrete set of values called quantized values, and (i,j) is a pair of integers. An example is a digital image of 512×512 pixels and each pixel takes on an integer value between 0 and 255. In this specification, an image shall mean a digital image unless otherwise noted.

A common way to represent the pixels of a digital color image is to use 3 numbers denoting the red component, green component, and blue component. In this way, the image is represented in the R-G-B color coordinate system. Another way to describe a color pixel is to use the luminance and 2 chrominance components, where the luminance corresponds to the brightness. There are many other color coordinate systems.

The process of representing an image by a stream of 'ones' and 'zeros', i.e. using bits, is commonly referred to as image coding. That is, an image is converted to a binary stream by image coding. Decoding refers to the process of obtaining the image from the binary stream. Image compression refers to the process of reducing the number of bits to represent a given image. In many image coding methods, it is desirable to use the fewest number of bits to represent the image. For this reason, image coding and image compression are often used synonymously.

Image coding or compression can be lossless or lossy. A lossless coding method produces a binary stream from which the original image can be obtained exactly. A lossy coding method produces a binary stream but the decoded image, called the compressed image, is not exactly the same as the original image. In lossy coding, the compressed image can look indistinguishable from the original, or it can look different from the original, in which case the difference shows up as artifacts.

Instead of being described by its pixel values, a digital image can also be described in the 'frequency domain' or more generally the 'transform domain'. The N×M values of f is transformed to a set of numbers called transform coefficients, usually also N×M in number. Commonly used transformations include the Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), and Wavelet Transform.

Current computer and information technology allow easy editing and perfect reproduction of digital images, which in turn can lead to problems with copyright protection, ownership verification and authentication. Such problems are addressed by digital watermarking, which concerns processes that embed or insert data into a multimedia data object. The inserted data are often called digital watermarks. Depending on the application, digital watermarking may be applied to different types of data, including digital still images, digital audio and digital video. For images, a visible watermark is one that is intentionally made to be noticeable to a human observer whereas an invisible watermark is one that is not perceptible to a human but may be extracted by a computer or other electronic means. Whether visible or invisible watermarking is employed depends upon the particular application. The following references may be consulted for further background on digital watermarking:

[1] Mintzer, et al., "Effective and Ineffective Digital Watermarks," IEEF-ICIP, 1997.

[2] Friedman, "The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image," IEEE Trans. on Consumer Electronics, November 1993.

[3] Schneider, et al., "A Robust Content Based Digital Signature for Image Authentication," IEEE-ICIP, 1996.

[4] Storck "A New Approach to Integrity of Digital Images," IFIP Conf. on Mobile Communication, 1996.

[5] Yeung, et al., "An Invisible Watermarking Technique for Image Verification," IEEE-ICIP, 1997.

[6] Swanson, et al., "Robust Data Hiding for Images,= IEEE DSP Workshop, 1996.

[7] Koch, et al., "Towards Robust and Hidden Image Copyright Labeling," IEEE Workshop on Nonlinear Signal and Image Processing, 1995.

[8] Zeng, et al., "On Resolving Rightful Ownership of Digital Images by Invisible Watermarks," IEEE-ICIP, 1997.

One application of digital watermarking is in the field of digital photography, in which images are captured with a digital camera or photographs are digitized. In these cases, it would be advantageous to embed an invisible watermark in the image at the time of capture or digitizing. This watermark could be used later (e.g., in a court of law) to verify that the image is authentic, i.e., has not been altered.

It is important that a method of waterinarking for authentication can:

(1) permit the user to determine whether an image has been altered or not;

(2) identify where in the image such alteration occurred; and

(3) allow the watermarked image stored in a lossy-compression format (such as JPEG).

In addition, it is highly desirable

(4) to integrate the watermark with the host image rather than as a separate data file; and

(5) to have the watermark invisible under normal viewing conditions.

Previously known methods for image authentication do not have all of the above capabilities. The digital signature methods (e.g., above cited references [2] [3] [4]) do not have capabilities 2 and 3; the pixel-domain watermarking methods (e.g., above cited reference [5]) do not have capability 3; the frequency-domain data hiding schemes (e.g., above cited [6] [7]) cannot always localize alterations and may introduce excessive distortion. Since the present invention and the pixel-domain method [5] have certain similarity, that method will be briefly reviewed and the difference pointed out.

The method presented in [5] embeds a watermark, which is a binary pattern, in the pixel domain, using a look-up table (LUT). The method is illustrated in FIG. 1, where the (unmarked) grey scale image consists of a block of 8×8 pixels whose values are shown and the pattern to be embedded is the letter "I", also formed with a block of 8×8 pixels. Suppose the black pixels of the pattern correspond to "0" and the white pixels correspond to "1". These "1"s and "0"s are called marking values. As shown in FIG. 1, the top row

of the LUT are the luminance values of the unmarked image and the bottom row are binary, i.e., "1" or "0". The 4th number in the first row of the image has a value of 153 as shown; the binary pattern corresponding to this pixel is black and therefore has the value "0". From the LUT, 153 corresponds to a "0", agreeing with the pixel value of the pattern. So the number 153 is unchanged in the marked image, i.e., the 4th pixel in the first row of the marked image has the value 153. The first number in the second row of the image has the value 144. The corresponding binary pattern is white and therefore has the value "1". But from the LUT, 144 corresponds to a "0". So the number 144 is changed to 143 for which the corresponding value in the table is "1 ". Thus, the first pixel in the second row of the marked image has the value 143. All pixels in the original image are processed in this manner. That is, if the luminance of a pixel in the original image does not map to the value in the corresponding binary pattern by the LUT, the luminance value is changed to a new value which is close to the original value and which corresponds to a binary value that agrees with the binary pattern.

The marked image is made up of an 8×8 block of pixels from which the watermark is easily extracted by referring to the LUT. The fourth pixel in the first row of the marked image is 153, for which the corresponding binary value from the LUT is "0". So the fourth pixel in the first row of the extracted pattern is black. Similarly, the first pixel in the second row of the marked image is 143, for which the corresponding binary value from the LUT is "1". So the first pixel in the second row of the extracted pattern is white. When all the pixels of the marked image have been processed in this manner, a pattern of "I" will have been extracted.

If a pixel in the marked image is changed, the changed value will be mapped to either "1" or "0", each with a probability of 0.5. So there is a 0.5 probability that the extracted watermark for that pixel will be different from the corresponding pixel in the original pattern. Such a possible change in a single pixel may or may not be observable by a viewer. However, if a group of neighboring pixels is changed, the probability that the corresponding part of the extracted watermark will look different is significantly increased. Images are often lossily compressed to save transmission time and storage. If an image is watermarked using the method just described, and if the marked image is then lossily compressed, the watermark inserted in the image will be changed due to the compression process. Therefore, the watermarked image from such an approach cannot be lossily compressed without adversely affecting the watermark. Another disadvantage of the above method is due to the fact that a human viewer can easily notice the changes in the pixel values in the smooth regions of the image due to the embedding process, making it difficult to insert the watermark in the smooth regions of an image.

As described in greater detail below, the present invention inserts a watermark in the quantized transform domain coefficients using a lookup table. This allows the watermarked image to be stored in compressed form. The present invention can also insert the pattern in the smooth regions of an image and can embed content based features in addition to a pattern, so the reliability of detecting alterations of the image is significantly increased. In addition, a shuffling scheme of the present invention can also be applied to embedding a watermark in the pixel domain of smooth regions.

The result achieved by present invention is illustrated with FIGS. 2A, 2B and 2C. FIG. 2A is a JPEG compressed image, into which we embed a watermark of the pattern "PUEE" (shown in FIG. 2B) using the method of the present invention. The watermarked image, shown in FIG. 2B, is indistinguishable from the unmarked original, FIG. 2A. Two modifications are then made of the image. "Princeton University" on the top right corner is changed to "Alexander Hall" and "Copyright 1997" in the lower left corner is changed to "January 1998", as illustrated in FIG. 2C. Also shown in FIG. 2C is that the watermark is extracted from this modified image, which clearly shows where modifications have taken place.

III. SUMMARY OF THE INVENTION

A primary object of the present invention is to provide a digital watermarking process whereby an invisible watermark inserted into a host image can be utilized to determine whether or not the image has been altered and, if so, where in the image such alteration occurred. This and other objects of the invention are achieved by the methods disclosed herein for applying a digital watermark to an image.

The inventive methods include the step of deriving from the image a plurality of component images, wherein each component image contains coefficients. The inventive methods employ at least one look-up table containing a plurality of coefficients and corresponding values, wherein at least some of the look-up table coefficients match coefficients in the component images. A watermark is then embedded in the image. The last step, embedding the watermark, is carried out by performing the following steps for at least some of the component images: (1) selecting a coefficient into which a marking value, representative of a corresponding portion of the watermark, is to be embedded; (2) using the value of the selected coefficient to identify a corresponding value in the look-up table(s); (3) leaving the identified coefficient unchanged if the corresponding value is the same as the marking value; and (4) changing the identified coefficient if the corresponding value is different from the marking value.

Another aspect of a preferred implementation of the invention is that, in the embedding step, the identified coefficient is changed minimally to a value having a corresponding value in the look-up table which is the same as the marking value. For example, the identified coefficient is preferably changed minimally by changing its value to that of the nearest coefficient having a corresponding look-up table value equal to the marking value.

Yet another aspect of the present invention concerns "shuffling" the coefficients prior to the embedding step. Such shuffling may involve concatenating the coefficients of a plurality of blocks into a string and randomly shuffling the order of the coefficients in the string. In this embodiment, the string is unshuffled after the embedding step.

Preferably, for error correction purposes, the marking value (e.g., marking bit) is embedded multiple times in each block, and a majority voting process is employed to decode the respective blocks.

The preferred embodiments also include the step of identifying selected coefficients as being unembeddable, such that the unembeddable coefficients are not employed to embed marking values. For example, DC coefficients (i.e., c_{00}) of the DCT process may be considered unembeddable. In addition, a threshold value may be selected and used such that coefficients having a value below the threshold value are considered small valued and thus unembeddable.

The preferred embodiments of the present invention can determine whether a marked image has been altered, and where such modification took place. In addition, after the

insertion of the watermark, the image may be stored in a lossy-compression form, thus permitting efficient storage and distribution. Moreover, the preferred embodiments produce two output signals for authentication: (1) a meaningful pattern to facilitate a quick visual check, and (2) an additional signal to detect unauthorized alteration. In addition, other information, such as content features, can be embedded into the image. The invention can be applied to an image compressed using JPEG or other techniques, such as Wavelet compression, and the marked image can be kept in the compressed format. Any alteration made on the marked image can be localized, making the preferred embodiment suitable for use in a "trustworthy" digital camera or camcorder. Furthermore, since the invention is computationally and structurally efficient (including the use of a look-up table), it can be applied in digital video authentication.

Other features and advantages of the present invention are described below.

IV. BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** depicts a pixel domain watermarking method of the above cited reference [5].

FIGS. **2A–2C** illustrate the overall result achieved by the present invention.

FIG. **3** depicts a JPEG still image compression method.

FIG. **4** illustrates an image and its wavelet coefficients.

FIG. **5** is a block diagram of a watermarking process of the present invention.

FIG. **6** depicts a transform domain embedding process employed by the present invention.

FIGS. **7A** and **7B** are block diagrams illustrating the embedding and authentication procedures, respectively, in accordance with the present invention.

FIGS. **8A** and **8B** are similar to FIGS. **7A** and **7B** in that they illustrate embodiments of the watermark embedding and authentication procedures, respectively, but include shuffling and inverse shuffling steps in accordance with another embodiment of the invention.

FIG. **9** depicts the shuffling process.

FIG. **10** illustrates a backup embedding scheme that may be employed in connection with smooth regions.

FIG. **11** illustrates three types of frames (I, P, B) in a video signal.

FIG. **12** illustrates a step in MPEG coding using motion compensation.

V. DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

We first illustrate the present invention by showing how to insert a binary pattern as the watermark in a JPEG compressed grey-scale image. The present invention, however, may be extended to other compression methods, such as Wavelet compression, to color images, and to video. Before describing the preferred embodiments of the invention, we will provide a brief overview of the JPEG method for lossy compression. Details can be found in the paper G. K. Wallace, "The JPEG still picture compression standard," IEEE Trans on Consumer Electronics, February 1992.

V.1 Overview of JPEG Lossy Compression

The JPEG method for image compression has been adopted as an international standard. When the image is lossily compressed, the compressed image may differ imperceptibly or significantly from the original image, depending on the parameters chosen to perform the compression.

The JPEG method is illustrated in FIG. **3**. To compress an image using the method, the image is first divided into blocks, typically of 8×8 pixels. Each block of pixels may be viewed as a component of the original image and hence be called a component image. For each block, a two-dimensional discrete-cosine-transform (DCT) is performed, resulting in 8×8 DCT coefficients for that block. The coefficients represent the spatial frequency components of the 8×8 block of pixels and are denoted by c_{ij}, where i and j range from 0 to 7. The c_{00} coefficient is called the DC coefficient, corresponding to the DC component of the original image block in both horizontal and vertical directions, while the c_{77} coefficient contains the highest frequency content of that block.

To lossily compress the image, each DCT coefficient is quantized, and the normalized quantized coefficients are run-length encoded, followed by Huffman or arithmetic encoding to produce a bit stream. To obtain the reconstructed image, the above process is reversed. That is, the normalized quantized coefficients are recovered from the bit stream, and inverse DCT of these coefficients taken to produce 8×8 pixels. All blocks of the 8×8 pixels thus obtained are put together to form the reconstructed image. If the quantized DCT coefficients differ very little from the original unquantized coefficients, then the reconstructed image differ imperceptibly from the original image. Otherwise, the reconstructed image will differ noticeably from the original image.

Many JPEG encoders use a quantization table to quantize the DCT coefficients by dividing each coefficient by the corresponding entry in the table and then rounding it to the nearest integer which is called the normalized quantized coefficient. To achieve different level of compression, the entries of a table is scaled up or down by a number called the quality factor Q. Table 1 together with a value of Q between 1 and 100 is often used for such quantization, as follows. For Q=50, the table is used unchanged as the quantization table. That is, each DCT coefficient is multiplied by the corresponding entry of Table 1 and the result is rounded and clipped. For Q>50, a new table for quantization is generated by multiplying Table 1 by (100−Q)/50. For Q<50, the new quantization table is generated by multiplying Table 1 by 50/Q. It should be mentioned that the present invention does not depend on this or any other quantization table.

TABLE 1

8 × 8 Quantization Matrix							
16	12	10	16	24	40	51	61
11	12	14	19	26	58	60	55
14	13	16	24	40	57	59	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	103
72	92	95	98	112	100	101	99

V.2 Subband and Wavelet Image Compression

Another well known image compression method is based on subbands. In this method, an image is filtered and the outputs of the filters are sub-sampled or down sampled. (To sub-sample by a factor of 2, one out of every two samples are deleted. To sub-sample by a factor of K, all but one from every K samples are deleted.) The results are component images, or more specifically highpass and lowpass component images if the filters used are highpass and lowpass filters. The component images usually bear certain resemblance to the original image. The process of filtering and

sub-sampling is often repeated several times. The final results are called subband coefficients, or subband images. These subband coefficients are then quantized and the quantized coefficients are then assigned to codes. In subband coding, the quantized coefficients from each subband are usually coded independently. Another well known method for image coding is based on wavelets, which is quite similar to subband coding in the initial steps. That is, the image is passed through a number of filters, and the outputs of these filters are subsampled to produce a number of coefficients, called wavelet coefficients.

Shown in FIG. **4** is an image and its wavelet coefficients. Notice the varying degree of resemblance of the seven components to the original image. The wavelet coefficients are then quantized and coded. Notice also that the seven components are not of the same size. The wavelet coefficients from one filter output is often correlated with those from other filter outputs, and wavelet coding takes advantage of such information. Although in principle this advantage can also be used in subband coding, what most researchers refer to as subband coding actually code each subband separately, as mentioned in the previous paragraph.

Details of these image coding methods can be found, for example, in M. Vetterli and J. Kovacevic, Wavelets and Subband Coding, Prentice Hall, 1995.

V.3 Overview of Preferred Embodiments

We first describe the present invention by demonstrating how to insert a pattern in a JPEG compressed grey-scale image. For simplicity, we shall use for illustration a binary pattern. We will discuss later how other information, such as content features, can be embedded into the image. The embedding of a binary pattern is done by modifying some of the normalized quantized DCT coefficients using the binary pattern and a look-up table (LUT) in such a way that the resulting marked image is visually not different from the original compressed image. The inserted pattern can be extracted from the marked image. The binary pattern can be a visually meaningful pattern such as a logo, letters, etc. Note that the marked image is in a compressed form. If the marked image is unchanged, then the extracted pattern will be identical to the original inserted pattern. If, on the other hand, the marked image was altered, then the extracted pattern will reveal where such alteration took place, as demonstrated earlier in FIG. **2**.

FIG. **5** schematically depicts a preferred embodiment of such a watermarking process in accordance with the present invention. As shown, an original image **10** undergoes a block by block DCT **20** and quantization **30**, which yields a set of normalized quantized coefficients **35**. The embedding process **40** modifies the quantization coefficients **35** to give a set of marked coefficients **45**. The embedding process **40** employs a prescribed set of data to be embedded **49** (i.e., the watermark) and a look-up table **60**, as described in detail below, to generate the marked coefficients **45**. The marked coefficients **45** are then input to an encoding process **70**, which yields a bit stream of the compressed watermarked image **80**. The inserted data can be extracted by simply reading from the LUT the value corresponding to the DCT coefficient of the marked image. If a part of the watermarked image is altered, the DCT coefficients from that part of image will be changed, hence the extracted watermark will differ from what was originally inserted, thus identifying where in the image the modification has occurred. This assumes that one bit of a binary watermark can be embedded in the corresponding block of the image. The question of how to make sure that it is possible to embed one bit in each block is discussed below.

V.4 Insertion of Binary Pattern Using a Look-up Table

We now discuss details of how to insert the watermark. As shown in FIG. **6**, a look-up table (LUT) **60**, which is generated beforehand, maps every possible value of JPEG coefficient to "1" or "0" with the lengths of consecutive "1" or "0" limited. Thus, the LUT **60** includes a row **60*a*** of JPEG coefficients and a row **60*b*** of corresponding "1"s and "0"s. To embed a "1" in one of the original JPEG coefficients **35**, that coefficient is unchanged if the entry of the table corresponding to that coefficient is also a "1". On the other hand, if the original coefficient corresponds to "0" in the table, that coefficient value is changed minimally to a neighboring value that has a "1" entry. The procedure for embedding of a "0" is similar, i.e., the coefficient is unchanged if its corresponding LUT entry is "0" but is changed minimally if its corresponding LUT entry is "1". These "1's" and "0's" are called marking values. It should be pointed out that the expression (to embed) as used herein does not necessarily require that the original data value be altered, but only that it is altered as necessary to make it equal to a value whose corresponding LUT value ("0" or "1") is equal to the value to be embedded.

In the example of FIG. **6**, zeros ("0"s) are to be embedded in two DCT coefficients of values "−73" and "24" respectively. Since the binary entry corresponding to "−73" is 1, "−73" is changed to "−74" for which the binary entry is "0" which is to be embedded. The coefficient value of "24" is unchanged because its binary entry is already "0".

The length of successive zeros and ones is limited to avoid noticeable distortion. The following procedures can be used to generate an L-entry look-up table T[.] with maximum allowed run of r.

Step-1: i=1.

Step-2: If i>r and T[i−1]=T[i−2]= . . . =T[i−r], then T[i]=1−T[i−1].

Otherwise, randomly select a number from {0, 1} and set T[i] to this value.

Step-3: Increase i by 1. If i>L, stop. Otherwise go back to Step-2.

Random numbers can be generated by pseudo-random generator; details can be found, for example, in R. Sedgewick, Algorithms in C, Addison-Wesley, 1990.

It should be noted, however, that there are many ways to generate such a table and the present invention is not limited to any one procedure. It should also be noted that more than one look-up table can be used to insert data and that the second entry of the table is not limited to be binary.

V.5 Selection of DCT Coefficients for Data Insertion

When the quantized coefficients are changed too much, the marked image will be visually different from the original image. On the other hand, a large change of the coefficients allows more information to be embedded into the image. The human visual system, in particular the just noticeable difference, has been used to determine how much can an image be changed without causing noticeable differences. The human visual system is explained, for example, in A. K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, 1989. The just noticeable differences is presented, for example, in A. J. Ahumada, et. al, "Luminance-model-based DCT quantization for color image compression," Proc. SPIE, Vol. 1666, 1992.

In accordance with the presently preferred implementation of the invention, several steps are taken to ensure that the markings are invisible. As mentioned above, the lengths of runs of "1" and "0" entries in the LUT is limited to avoid excessive modification on the coefficients. The DC coefficients is unchanged unless the quantization step is small.

Also, the small valued AC coefficients are unchanged. Those DCT coefficients that cannot be changed because of these constraints are called unembeddable, and those coefficients that can be changed are called embeddable. On the average, between 6 to 10 of the coefficients in a block of a typical natural image are embeddable. This means one can normally embed more than one bit of information in each block. All the embeddable coefficients in a block can be used to embed the same bit in the binary pattern in the manner described in the previous section. In this case, the extraction of the embedded information can be done using majority voting. The reason for using more than one coefficients to embed same information is explained in the next section. However, the number of embeddable coefficients vary significantly from block to block and some blocks may not contain any embeddable coefficients. In particular, it is difficult to embed data in smooth blocks where all AC coefficients are very small and hence unembeddable. If no bits of the watermark is embedded in a block, then the alteration of that block would be difficult to detect. To over come this difficulty, two methods are presented later in the section entitled Smooth Region Embedding.

V.6 Embed Other Data

The watermark that was inserted in the previous discussion is a binary pattern which often can provide a quick visual way of detecting modifications. It is clear that a non-binary pattern can be similarly embedded. It is also possible to embed 'content features'. A simple example of such features is the most significant bit of average intensity of a block or of a group of blocks such as a macroblock consisting of 4 blocks (16×16 pixels). The combination of patterns and features is suitable for such applications as image authentication for "trustworthy" digital cameras. The embedding and authentication procedures are illustrated in FIGS. 7A and 7B.

As shown in FIG. 7A, the embedding process is the same as the basic process illustrated in FIG. 5 except that content features are extracted from the original image **10** or from the DCT coefficients **25** or from both. The content features **90** are input to the embedding process **40** along with the pattern **50**.

The authentication process, as shown in FIG. 7B, starts with a test image **100** (i.e., a watermarked image whose authenticity is to be tested). It is transformed by a block DCT process **20** and then quantized in the same manner as in the embedding/watermarking process to produce normalized quantized coefficients **36**. Content features **91** are also extracted from the test image. The watermark is extracted by an extraction process **110**, which is the inverse of the embedding process **40**. In the extraction process **110**, the normalized quantized DCT coefficients **36** are examined in view of the look-up table **60** (the same table as used in the watermarking process of FIGS. 5 and 7A). The outputs of the extract process **110** are a pattern **51** and the features **92**. The pattern **51** can be visualized by using visualize process **120**, such as a printer or computer monitor, to allow a human observer to quickly identify which part of the image has been changed. The extracted pattern **51** can also be compared with the original pattern **50**, if available, to detect possible alterations. The compare process **130** compares the extracted features **92** with the features **91** to provide an additional detection of possible alterations. If the test image **100** is the same as the marked image without modification and if the bit stream of the test image is used for authentication, its normalized quantized coefficients are available from the bit stream and are the same as the coefficients **35**, and the features **90**, **91** and **92** are identical.

If the test image **100** is the same as the marked image without modification but it is presented for authentication as an image, then a small number of coefficients **36** may be different from coefficients **35** because of the rounding error in the computation of coefficients. Error correction coding, such as using majority voting as mentioned in the previous section, can be incorporated in the embed process **40** to guard against such errors.

V.7 Smooth Region Embedding

The percentage of smooth regions in an image varies widely, depending on the subject matter. But typically, 20% of a natural image may be regarded as smooth. As mentioned previously, blocks in the smooth region of an image may not have any DCT coefficient to insert the watermark or to embed features. This is true in spite of the fact that between 10 to 15% of the DCT coefficients in a natural image are embeddable. Thus, on the average, there are 7 to 10 embeddable coefficients per block on average. In other words, there are plenty of embeddable coefficients in an image, but the uneven distribution of them leaves some blocks with no coefficients in which to embed data.

This difficulty can be circumvented by the following shuffling process. For an image divided into blocks of 8×8 pixels, the quantized DCT coefficients of all blocks can be concatenated into a single string, of which the first 64 numbers are from the first image block, the next 64 numbers are from the second image block, and so forth. Some of the groups of 64 coefficients may not have any coefficient that are embeddable. Suppose the order of this string is randomly shuffled to produce a second string. Each coefficient in the second string comes from one and only one coefficient in the first stream, that is, before shuffle. Because of the random shuffle, the probability that each group of 64 numbers in the second string contains at least one embeddable coefficient is significantly increased. In fact, it can be calculated that the probability of no embeddable coefficients in a group of 64 coefficients in the second string is about 0.1%. Therefore, the coefficients in the shuffled string can be used to embed data as described previously, even though these 64 coefficients are not those computed from that block.

The embedding process when shuffling is used is illustrated in FIG. 8A. It is identical to that described above except that the quantized DCT coefficients of all blocks are concatenated into a single string and randomly shuffled, in the process **140**, before embedding. Suppose the blocks are of size 8×8. Then each of the 64 coefficients of the shuffled string are used to embed the data for one block.

This procedure is illustrated with the help of FIG. 9 for the case of embedding a binary pattern. Suppose the coefficients of the first block have values: 171 165 144 . . . , the coefficients for the second block have values: 180 201 192 . . . , the coefficients of the third block have values: 130 125 128 . . . , etc., as shown in the first sequence. The shuffling reordered this sequence and produces the second sequence. In this illustration, the first block of the shuffled coefficients have values: 180 165 125 130 . . . , where the first coefficient "180" comes from the first coefficient of block 2 in the original (unshuffled) sequence, the second coefficient "165" is not moved by the shuffle, the third coefficient "125" comes from the second coefficient of the original block 3, and the fourth coefficient "130" comes from the first coefficient of the original block 3. Suppose a "1" is to be embedded in block one and suppose this embedding changes the second coefficient "165" to "164" as shown in the third sequence of FIG. 9. After embedding, all coefficients are inverse shuffled, which is the fourth sequence of FIG. 9. It should be clear that the embedding of features can be done identically.

FIG. 8B depicts the authentication process for a test image **100**. It is transformed by a block DCT process **20**, quantized by the process **30**, and then shuffled by the process **140**, in the same manner as in the embedding process. The watermark is extracted by an extraction process **110**. Using the LUT **60**, the extract process **111** extracts the embedded pattern **53** and the embedded features **93**. The extracted pattern **53** is visualized by the process **122** and the features **93** are compared with features **91** by the compare process **132**. The visualize process **122** may be different from the visualize process **120** (FIG. **7**) and the compare process **132** may be different from the compare process **130** (FIG. **7**) because processes **122** and **132** may include inverse shuffle and other operations.

It should be noted that in addition to the random shuffling involving all the coefficients in one shuffle, other reordering of the DCT coefficients is possible, such as dividing all the coefficients of the image into several parts and shuffling each part separately.

It is mentioned above that the pixel domain embedding process may encounter difficulty when embedding data in the smoothed region of the image. The shuffling scheme described can be used to embed data in the pixel domain in the smooth regions of the image. As a simple example, suppose we embed one bit in every four pixels. Each pixel in the image is determined to be in a smooth region or not. Those in the smooth region are labeled unembeddable, and those not in the smooth region are embeddable. All pixels are then randomly shuffled. With the shuffle, the probability that any four pixels will have at least one embeddable pixel is significantly increased. The data is embedded and the pixels are inverse shuffled after the embedding.

Another way of smooth region embedding is to embed data in blocks that are not smooth but whose location bears a fixed relationship to that of the smooth block in question. This method is simple to implement, as illustrated in FIG. **10**. As shown, instead of embedding one bit independently in each block (8x8 pixels), we take a macroblock (16x16 pixels) as a unit, and use two bits to embed data corresponding to the current macroblock which is not smooth, and use the other two bits to embed the data corresponding to the companion macroblock which is smooth. That is, we use companion macroblocks as a backup. In the illustration of FIG. **10**, the companion macroblocks are separated by half the image height, i.e., the two companion macroblocks are at location (i,j) and (i,j+H/2), where H is the image height in terms of number of blocks. One can also group 4 macroblocks located, say at (i,j), (i,j+H/2), (i+W/2,j) and (i+W/2, j+H/2) as companion macroblocks, where W is the image width in terms of number of blocks.

V.8 Detection of Alteration

As described above, modifications on a watermarked image are detected by visualizing the extracted pattern, or by comparing the pattern with the original pattern, or by comparing the extracted features using the LUT with the features of the test image. If the watermarked image has been changed, then the DCT coefficients from that part of the image that have been modified will be changed, hence the extracted watermark will be changed. Suppose one bit from the watermark is inserted into each block. The watermark extracted from the block in question after alteration has a 50% probability to be a "1" or a "0", hence there is 50% probability that the extracted bit may disagree with the bit originally inserted before the alteration. Since intentional modification of the image usually involves regions consisting of more than one block, this 50% detection probability is increased to 75% if two blocks are changed, to 87.5% if three blocks are changed, and to 93.75% if four blocks are changed.

Furthermore, suppose a block has more than one embeddable coefficients and we embed the same bit in all these coefficients. Suppose this block is modified and there are now N embeddable coefficients after modification. The probability that the extracted watermark bit from each coefficient is "1" or "0" each with a probability of 0.5. So the probability of all extracted bits agreeing with the originally inserted bit is reduced to (0.5) N. In general, the probability of correct detection of image modification will be increased if more than one bit per block are inserted into the marked image.

If shuffling is used, the detection of alteration can be illustrated by the following simple example. Suppose one block was altered. The embeddable coefficients from this block are now in different shuffled blocks, and the extracted bits from each of these coefficients have 0.5 probability of being changed. Suppose this is the only block that has been altered. Then error correction coding from the bits extracted from the shuffled blocks will identify those coefficients that have been changed. These coefficients would have to come from the altered block. The 'content features' embedded as watermarks can also be used to detect alteration.

V.9 Deterring Forgery of Watermarks

If the same look-up-table and the same pattern are used to mark many images, it is possible to obtain information about the table and the pattern from these images. Thus, to deter forgery, it is important to deter any attempt obtain information about the table and the pattern from the images. A simple way do this is to modulate the bits to be inserted by a pseudo random sequence. Depending on the computation and memory resources available for the particular application, such a pseudo random sequence can be either a fixed one or one selected from a list of such sequences. The shuffling process as described previously also helps to deter such an attempt, because it is difficult to forge watermarks without prior knowledge of the shuffling.

VI. Extensions

VI.1 Double Watermarking for Authentication and Ownership Verification

Previous work on double watermarking mainly emphasized embedding multiple labels (such as an ownership label, recipient label, etc.) using the same embedding approach. The present invention may be used in combination with watermarking methods for ownership verification such as described in the above cited reference [8].

VI.2 Color Images

Whereas each pixel in a grey scale image is represented by a single number, each pixel in a color image is represented by more than one numbers, such as the three components in the RGB or the YCrCb color coordinates. The present invention can be applied separately to each of the coordinates or to combination of coordinates.

VI.3 Other Compression Methods

The present invention can be applied to images compressed using methods other than JPEG. In wavelet compression, for example, the image is transformed into a number of component images from which the original image can be reconstructed. The component images are quantized. The present invention can be used to insert a watermark or watermarks into the component images by modifying the quantized coefficients using one or more look-up tables.

VI.4 Multi-level Data Embedding and Unequal Error Protection

As mentioned above, two sets of data, namely, a meaningful visual pattern and a set of low-level content features, may be embedded in the image for authentication purposes. Multilevel data embedding and unequal error protection can

be employed to embed several sets of data with different error protection capabilities in accordance with their importance.

VI.5 Video Coding and Authentication of Video

A video consists of many images arranged in an orderly sequence. The individual images are called frames or pictures. Some video systems also divide each frame into two fields, called the even field and the odd field. Digital video refers to a video sequence in which the individual frames have been digitized.

Video coding or video compression relies on the information redundancies found within each frame and among frames. A simple way to code a digital video is to code each frame separately, such as in the method known as motion JPEG. In the well known methods of MPEG-1 and MPEG-2, the frames are divided into three types: the intra-coded or I-frames, the predictive-coded or P-frames, and the bi-directional coded or B-frames. A typical MPEG coded video may have a pattern such as I B B P B B P B . . . , as illustrated in FIG. 11. The I-frames are codes by themselves, similar to JPEG coding. Each P-frame is divided into blocks, typically of size 8×8 pixels. Each block of pixels is matched with the pixels in the previous I or previous P-frame and the best 8×8 pixels in that I- or P-frame is called the reference block. A block in a P-frame and its reference block in a previous I- or P-frame is illustrated in FIG. 12. The difference is taken between each block and the reference block. This difference, called the residue or the displaced-frame-difference, is coded using a JPEG like method. The relative position of the block in the current P-frame and that of the reference block is called the motion vector of the block. Motion vectors are often determined using 4 blocks together (16×16 pixels), called a macroblock. Motion compensation coding refers to the determination of motion vectors and the coding of the residues.

The coding of B-frames is identical to that of P-frames except each block in the B-frame in question is matched with the I-frames and P frames either in the backward or the forward directions, or both. As for the P-frames, the residue after motion compensation of the B-frames is coded.

The present invention can be applied to digital video authentication. Consider, for the purpose of discussion, digital video compressed using the MPEG-1 and MPEG-2 standards. Since the I-frame is compressed in much the same way as a JPEG still image, the present invention can be applied straightforwardly.

To watermark P-frames that are coded using motion compensation, one can begin with a digital video where each frame has been watermarked using the approach described above. After MPEG compression, the watermark embedded in the I-frames can be extracted as discussed above. The extracted watermark embedded into the P-frames before MPEG encoding may be distorted from that originally inserted. If this is the case, then the residue after motion compensation, called the displaced frame difference, is modified. This process is repeated until the embedded watermark can be reliably extracted.

The true scope of the present invention is not limited to the presently preferred embodiments disclosed above. For example, the invention is not limited to processes employing the DCT, or to processes in which 8×8 or even 16×16 blocks of data are used to embed a "0" or "1 " bit of a watermark. Other possible modifications of the preferred embodiments will be apparent to those skilled in the art.

We claim:

1. A method for applying a digital watermark to an image, comprising the steps of:

(A) deriving from the image a plurality of component images, wherein each component image contains coefficients;

(B) providing at least one look-up table containing a plurality of coefficients and corresponding values, wherein at least some of the look-up table coefficients match coefficients in said component images;

(C) embedding said watermark in said image by performing the following substeps for at least some of said component images: (1) selecting a coefficient into which a marking value, representative of a corresponding portion of the watermark, is to be embedded; (2) using the value of the selected coefficient to identify a corresponding value in said look-up table(s); (3) leaving the selected coefficient unchanged if the corresponding value is the same as the marking value; and (4) changing the selected coefficient if the corresponding value is different from the marking value;

further comprising the step of identifying selected coefficients as being unembeddable, wherein said unembeddable coefficients are not employed in step (C) to embed marking values, and wherein some of said coefficients are identified as DC coefficients, and said DC coefficients are considered unembeddable; and

further comprising selecting a threshold value, wherein coefficients having a value below said threshold value are considered small valued and thus unembeddable.

2. A method for applying a digital watermark to an image, comprising the steps of:

(A) deriving from the image a plurality of component images, wherein each component image contains coefficients;

(B) providing at least one look-up table containing a plurality of coefficients and corresponding values, wherein at least some of the look-up table coefficients match coefficients in said component images; and

(C) embedding said watermark in said image by performing the following substeps for at least some of said component images: (1) selecting a coefficient into which a marking value, representative of a corresponding portion of the watermark, is to be embedded; (2) using the value of the selected coefficient to identify a corresponding value in said look-up table(s); (3) leaving the selected coefficient unchanged if the corresponding value is the same as the marking value; and (4) changing the selected coefficient if the corresponding value is different from the marking value;

further comprising the step of identifying selected coefficients as being unembeddable, wherein said unembeddable coefficients are not employed in step (C) to embed marking values; and

further comprising selecting a plurality of threshold values, wherein coefficients having values below said threshold values are considered small valued and thus unembeddable.

3. A method for applying a digital watermark to an image, comprising the steps of:

(A) deriving from the image a plurality of component images, wherein each component image contains coefficients;

(B) providing at least one look-up table containing a plurality of coefficients and corresponding values, wherein at least some of the look-up table coefficients match coefficients in said component images;

(C) shuffling the coefficients prior to said embedding step; and

(D) embedding said watermark in said image by performing the following substeps for at least some of said component images: (1) selecting a coefficient into which a marking value, representative of a corresponding portion of the watermark, is to be embedded; (2) using the value of the selected coefficient to identify a corresponding value in said look-up table(s); (3) leaving the selected coefficient unchanged if the corresponding value is the same as the marking value; and (4) changing the selected coefficient if the corresponding value is different from the marking value;

wherein the image is divided into a plurality of component images of n×m pixels, wherein n and m are integers, and wherein each component image is transformed using the discrete cosine transform (DCT) into n×m transform coefficients; wherein the corresponding values in said look-up table(s) are binary-valued and are constrained to have a prescribed maximum run length of either binary value; wherein, in the embedding substep (D)(4), the selected coefficient is changed minimally to a value having a corresponding value in the look-up table(s) whose value is the same as the marking value, wherein the selected coefficient is changed minimally by changing its value to that of the nearest coefficient having a corresponding look-up table value equal to the marking value.

4. A method as recited in claim 3, wherein, prior to the embedding step (C), the image is compressed using JPEG compression.

5. A method as recited in claim 3, further comprising the step of shuffling the coefficients prior to said embedding step, wherein said shuffling step comprises concatenating the coefficients of a plurality of component images into a string and shuffling the order of the coefficients in said string; and wherein, after embedding, the string is inverse shuffled.

6. A method as recited in claim 3, wherein the marking value is embedded multiple times in each component image, whereby a majority voting process may be employed to decode the respective component images.

7. A method as recited in claim 3, wherein the marking value is embedded using error correction encoding, and wherein error correction decoding is employed to extract the marking values.

8. A method as recited in claim 7, wherein the error correction encoding comprises embedding of the marking value multiple times and the decoding comprises the use of majority voting.

9. A method as recited in claim 3, wherein the marked image is decodable to determine whether it has been altered and, if so, where in the image such alterations were made.

10. A method as recited in claim 3, further comprising the step of storing the marked image in a lossy-compression form.

11. A method as recited in claim 3, wherein the method is used in a digital camera or camcorder.

12. A method for applying a digital watermark to an image, comprising the steps of:

(A) deriving from the image a plurality of component images, wherein each component image contains coefficients;

(B) providing at least one look-up table containing a plurality of coefficients and corresponding values, wherein at least some of the look-up table coefficients match coefficients in said component images;

(C) shuffling the coefficients; and

(D) embedding said watermark in said image by performing the following substeps for at least some of said component images: (1) selecting a coefficient into which a marking value, representative of a corresponding portion of the watermark, is to be embedded; (2) using the value of the selected coefficient to identify a corresponding value in said look-up table(s); (3) leaving the selected coefficient unchanged if the corresponding value is the same as the marking value; and (4) changing the selected coefficient if the corresponding value is different from the marking value;

wherein the image is divided into a plurality of component images of n×m pixels, wherein n and m are integers, and wherein each component image is transformed using the discrete cosine transform (DCT) into n×m transform coefficients; wherein the corresponding values in said look-up table(s) are binary-valued and are constrained to have a prescribed maximum run length of either binary value; wherein, in the embedding substep (C)(4), the selected coefficient is changed minimally to a value having a corresponding value in the look-up table(s) whose value is the same as the marking value, wherein the selected coefficient is changed minimally by changing its value to that of the nearest coefficient having a corresponding look-up table value equal to the marking value; and

further comprising the step of identifying selected coefficients as being unembeddable, wherein said unembeddable coefficients are not employed in step (C) to embed marking values; wherein some of said coefficients are identified as DC coefficients that are considered unembeddable; and wherein coefficients having values below threshold values are considered unembeddable.

13. A method for applying a digital watermark to an image, comprising the steps of:

(A) deriving from the image a plurality of component images, wherein each component image contains coefficients;

(B) providing at least one look-up table containing a plurality of coefficients and corresponding values, wherein at least some of the look-up table coefficients match coefficients in said component images;

(C) shuffling the coefficients prior to said embedding step; and

(D) embedding said watermark in said image by performing the following substeps for at least some of said component images: (1) selecting a coefficient into which a marking value, representative of a corresponding portion of the watermark, is to be embedded; (2) using the value of the selected coefficient to identify a corresponding value in said look-up table(s); (3) leaving the selected coefficient unchanged if the corresponding value is the same as the marking value; and (4) changing the selected coefficient if the corresponding value is different from the marking value.

wherein the image is divided into a plurality of component images of size n×m pixels, wherein n and m are integers, and wherein each component image is transformed into coefficients;

further comprising the step of shuffling the coefficients prior to said embedding step;

wherein said shuffling step comprises concatenating the coefficients of a plurality of component images into a string and shuffling the order of the coefficients in said string; and

wherein the shuffled string of coefficients are put into a plurality of sub-strings and each sub-string is used to

embed one watermark or is to be used for error detection or for error correction.

14. A method for applying a digital watermark to an image, comprising the steps of:

(A) deriving from the image a plurality of component images, wherein each component image contains coefficients;

(B) providing at least one look-up table containing a plurality of coefficients and corresponding values, wherein at least some of the look-up table coefficients match coefficients in said component images;

(C) shuffling the coefficients prior to said embedding step; and

(D) embedding said watermark in said image by performing the following substeps for at least some of said component images: (1) selecting a coefficient into which a marking value, representative of a corresponding portion of the watermark, is to be embedded; (2) using the value of the selected coefficient to identify a corresponding value in said look-up table(s); (3) leaving the selected coefficient unchanged if the corresponding value is the same as the marking value; and (4) changing the selected coefficient if the corresponding value is different from the marking value.

wherein the image is divided into a plurality of component images of size n×m pixels, wherein n and m are integers, and wherein each component image is transformed into coefficients;

further comprising the step of shuffling the coefficients prior to said embedding step;

wherein said shuffling step comprises concatenating the coefficients of a plurality of component images into a string and shuffling the order of the coefficients in said string; and

wherein the shuffled string of coefficients are put into a plurality of sub-strings and each sub-string is used to embed one watermark or is to be used for error detection or for error correction.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 6,285,775 B1

DATED : September 4, 2001

INVENTOR(S) : Min Wu et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 2,
Line 18, delete "=" and insert -- " -- therefor.
Line 34, delete "waterinarking" and insert -- watermarking -- therefor.

Column 10,
Line 45, delete "arc" and insert -- are -- therefor.

Column 12,
Line 27, insert "to" after the word "attempt".
Line 49, delete "numbers" and insert -- number -- therefor.

Signed and Sealed this

Seventh Day of May, 2002

Attest:

JAMES E. ROGAN

Attesting Officer

Director of the United States Patent and Trademark Office

US006064764A

United States Patent [19]

Bhaskaran et al.

[11] Patent Number: 6,064,764

[45] Date of Patent: May 16, 2000

[54] **FRAGILE WATERMARKS FOR DETECTING TAMPERING IN IMAGES**

[75] Inventors: **Vasudev Bhaskaran**, Mountain View; **Viresh Ratnakar**, Sunnyvale, both of Calif.

[73] Assignee: **Seiko Epson Corporation**, Tokyo, Japan

[21] Appl. No.: **09/052,041**

[22] Filed: **Mar. 30, 1998**

[51] **Int. Cl.**^{7} **G06K 9/00**

[52] **U.S. Cl.** **382/183**; 382/100

[58] **Field of Search** 382/100, 183, 382/232, 235, 238, 239, 243, 244, 245, 246; 358/426, 261.1, 261.2, 261.3, 427, 261.4, 428, 430, 431, 432; 713/176; 380/54, 10, 232

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,488,664	1/1996	Shamir .	
5,530,759	6/1996	Braudaway et al. .	
5,568,570	10/1996	Rabbani .	
5,606,609	2/1997	Houser et al. .	
5,613,004	3/1997	Cooperman et al. .	
5,617,119	4/1997	Briggs et al. .	
5,664,018	9/1997	Leighton .	
5,687,236	11/1997	Moskowitz et al. .	
5,689,587	11/1997	Bender et al. .	
5,699,427	12/1997	Chow et al.	380/3
5,809,139	9/1998	Girod et al.	380/5
5,822,436	10/1998	Rhoads	380/54
5,825,892	10/1998	Braudaway et al.	380/51
5,848,155	12/1998	Cox	380/4
5,862,217	1/1999	Steinberg et al. .	
5,862,218	1/1999	Steinberg .	
5,875,249	2/1999	Mintzer et al.	380/54
5,915,027	6/1999	Cox et al.	380/54
5,930,369	7/1999	Cox et al.	380/54
5,942,414	8/1999	Cass et al.	382/183
5,960,081	9/1999	Vynne et al.	380/10
5,991,426	11/1999	Cox et al.	382/100

OTHER PUBLICATIONS

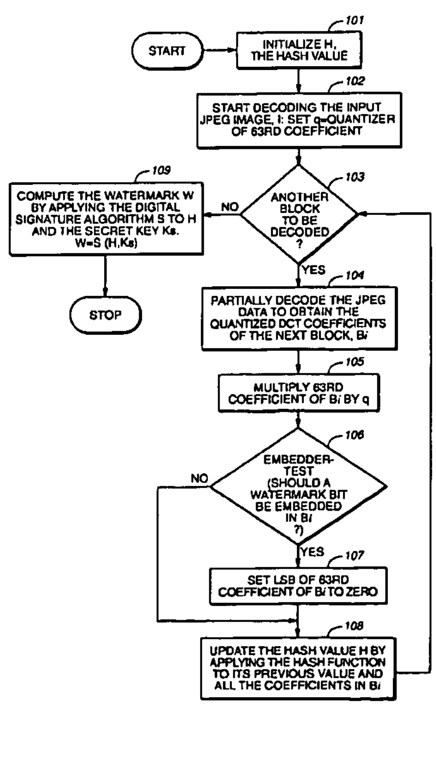
"*A Two-Dimensional Digital Watermark*", A.Z. Tirkel, et al. Pag 1–7 Date Sep. 1988.

Primary Examiner—Bisan Tadayon
Assistant Examiner—Seyed Azarian

[57] **ABSTRACT**

A watermarking scheme for images which includes techniques for inserting and extracting fragile watermarks in the frequency domain and for determining whether an image so watermarked has been tampered with. Watermark insertion is accomplished by embedding the bits of a digital signature of a hash function of the image in the frequency coefficients of the image. Tamper detection is accomplished generally as follows: the fragile watermark which was embedded during the watermark insertion process is extracted from the image; the hash function of the image is computed as in the insertion process; it is verified using a public key whether the extracted watermark is a valid signature of the hash value. If so, then there is assurance that the image has not been tampered with. Otherwise, there is reason to conclude that the image has been tampered with.

21 Claims, 5 Drawing Sheets



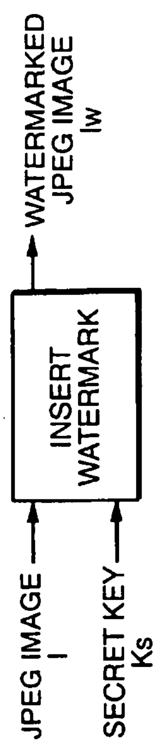


FIG._1

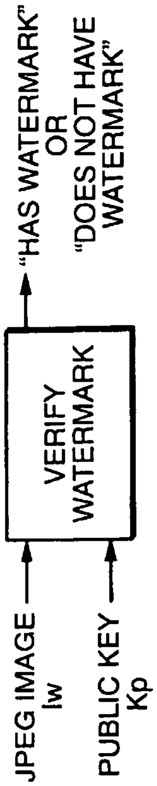


FIG._4

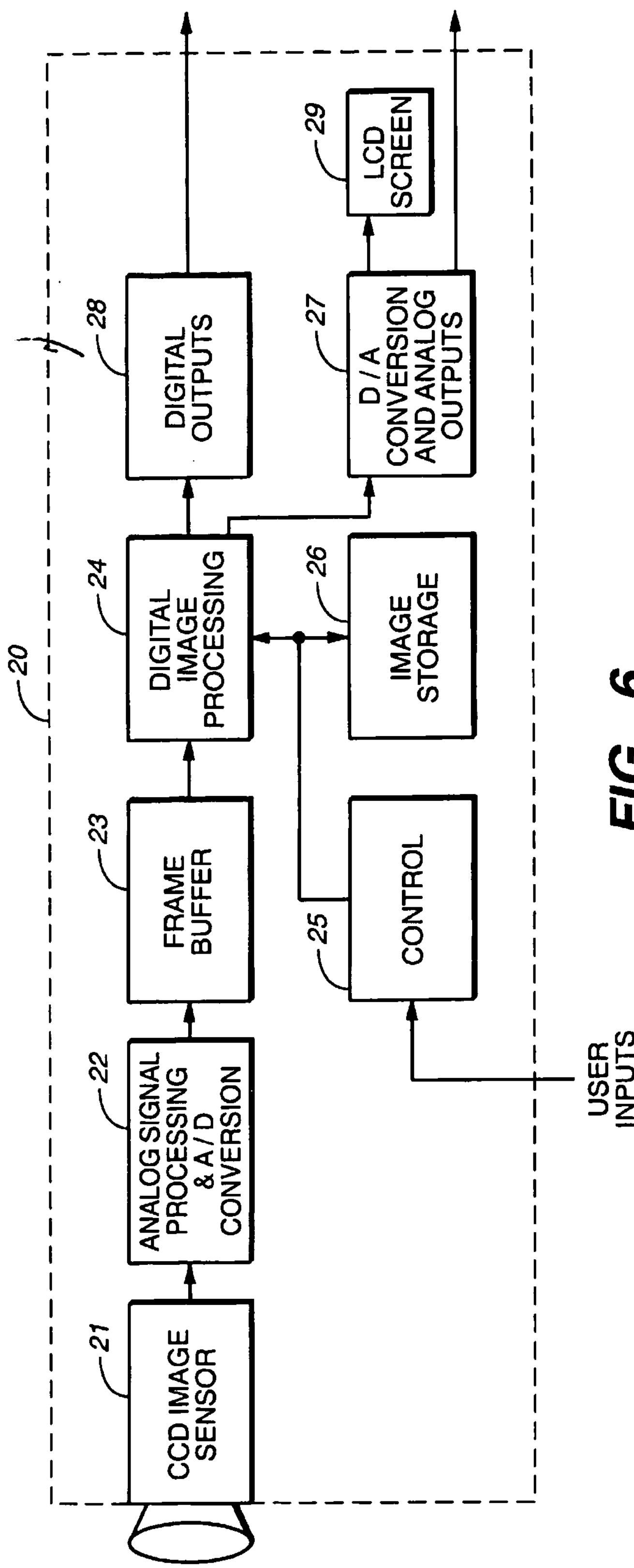


FIG._6

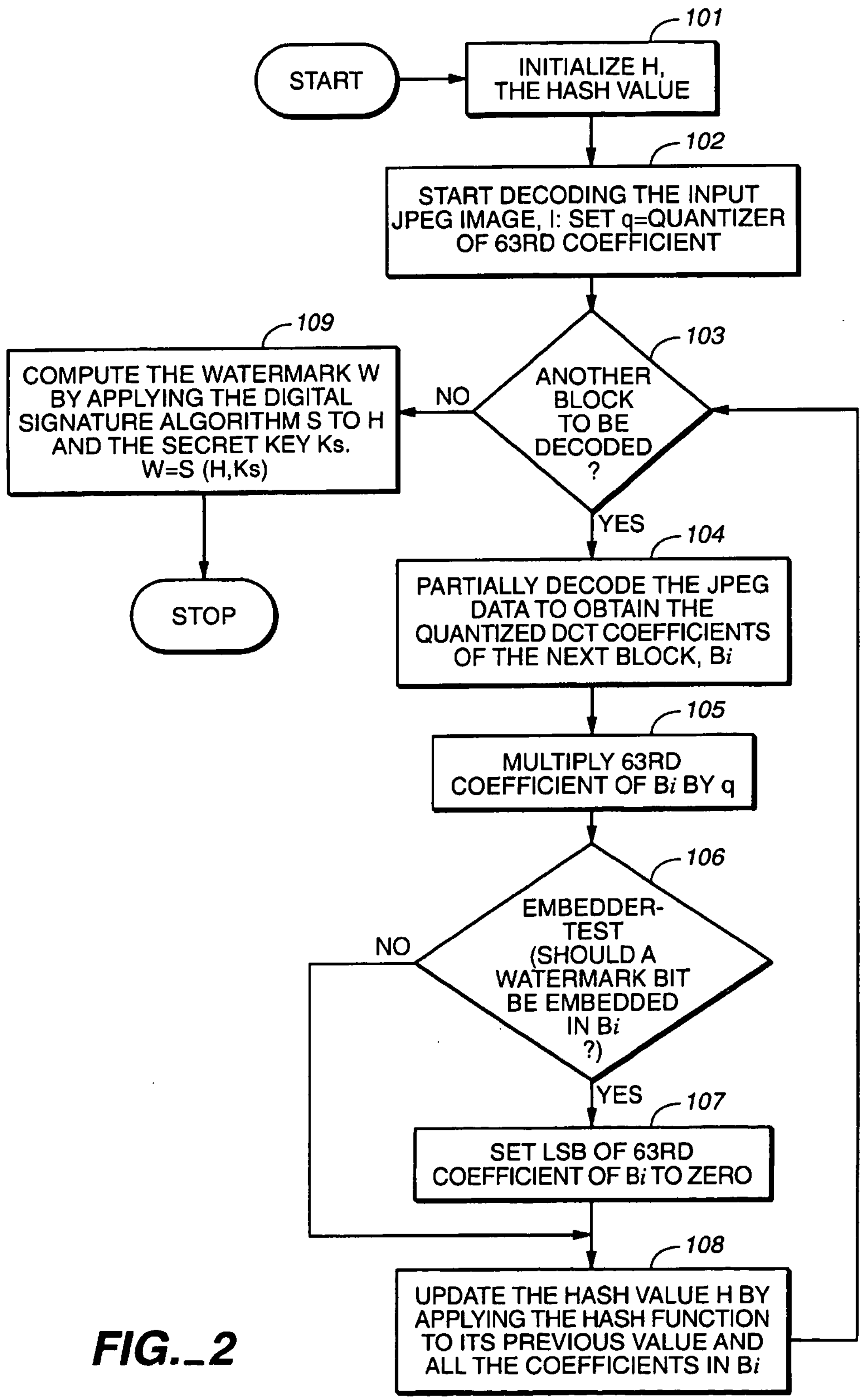
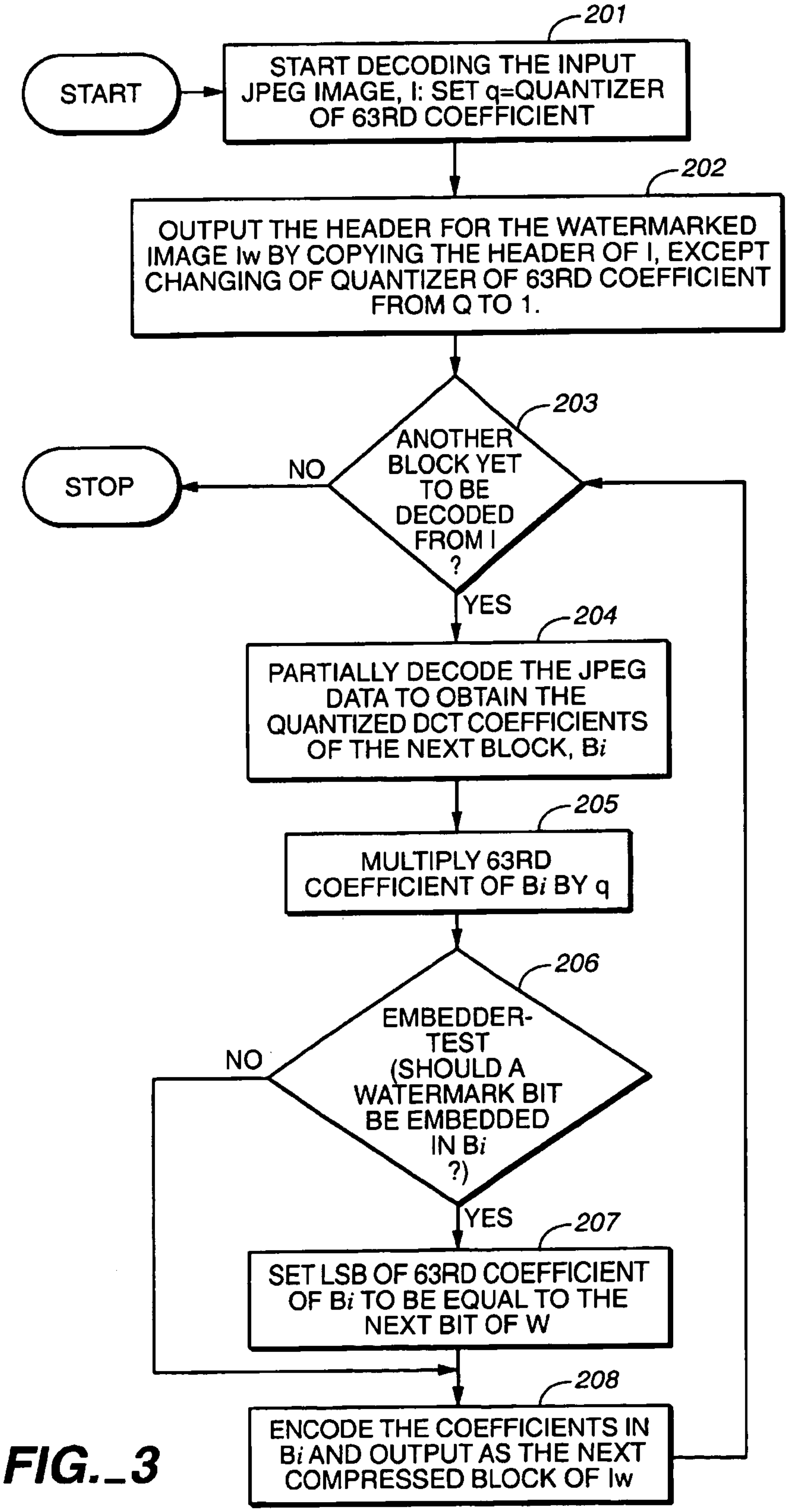

START
101
INITIALIZE H, THE HASH VALUE
102
START DECODING THE INPUT JPEG IMAGE, I: SET q=QUANTIZER OF 63RD COEFFICIENT
103
ANOTHER BLOCK TO BE DECODED ?
NO
109
COMPUTE THE WATERMARK W BY APPLYING THE DIGITAL SIGNATURE ALGORITHM S TO H AND THE SECRET KEY Ks. W=S (H,Ks)
STOP
YES
104
PARTIALLY DECODE THE JPEG DATA TO OBTAIN THE QUANTIZED DCT COEFFICIENTS OF THE NEXT BLOCK, Bi
105
MULTIPLY 63RD COEFFICIENT OF Bi BY q
106
EMBEDDER-TEST (SHOULD A WATERMARK BIT BE EMBEDDED IN Bi ?)
NO
YES
107
SET LSB OF 63RD COEFFICIENT OF Bi TO ZERO
108
UPDATE THE HASH VALUE H BY APPLYING THE HASH FUNCTION TO ITS PREVIOUS VALUE AND ALL THE COEFFICIENTS IN Bi

FIG._2



201
START
START DECODING THE INPUT JPEG IMAGE, I: SET q=QUANTIZER OF 63RD COEFFICIENT
202
OUTPUT THE HEADER FOR THE WATERMARKED IMAGE Iw BY COPYING THE HEADER OF I, EXCEPT CHANGING OF QUANTIZER OF 63RD COEFFICIENT FROM Q TO 1.
203
ANOTHER BLOCK YET TO BE DECODED FROM I ?
NO
STOP
YES
204
PARTIALLY DECODE THE JPEG DATA TO OBTAIN THE QUANTIZED DCT COEFFICIENTS OF THE NEXT BLOCK, Bi
205
MULTIPLY 63RD COEFFICIENT OF Bi BY q
206
EMBEDDER-TEST (SHOULD A WATERMARK BIT BE EMBEDDED IN Bi ?)
NO
YES
207
SET LSB OF 63RD COEFFICIENT OF Bi TO BE EQUAL TO THE NEXT BIT OF W
208
ENCODE THE COEFFICIENTS IN Bi AND OUTPUT AS THE NEXT COMPRESSED BLOCK OF Iw

FIG._3

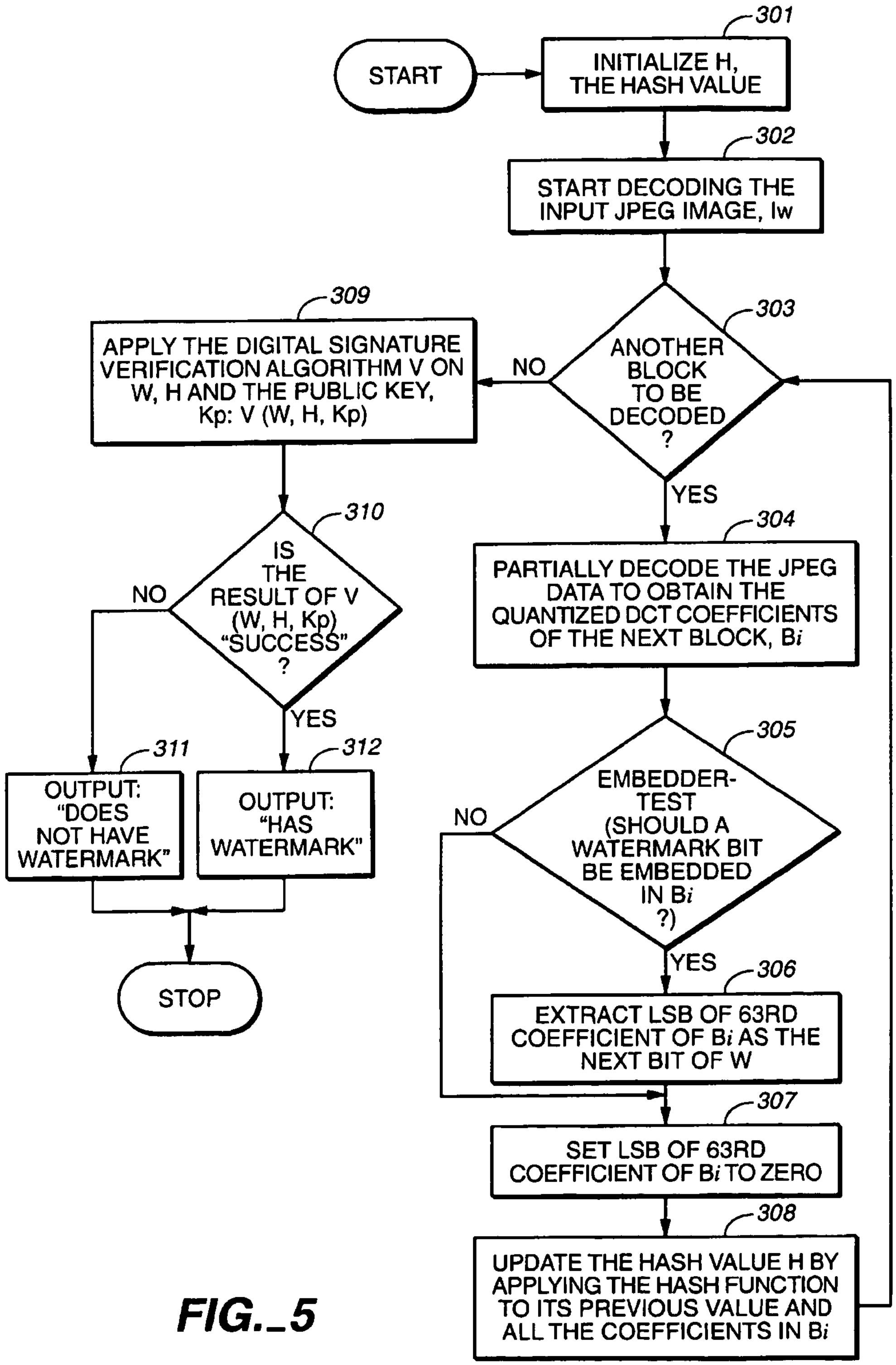

START
301
INITIALIZE H, THE HASH VALUE
302
START DECODING THE INPUT JPEG IMAGE, Iw
303
ANOTHER BLOCK TO BE DECODED ?
NO
YES
304
PARTIALLY DECODE THE JPEG DATA TO OBTAIN THE QUANTIZED DCT COEFFICIENTS OF THE NEXT BLOCK, Bi
305
EMBEDDER-TEST (SHOULD A WATERMARK BIT BE EMBEDDED IN Bi ?)
NO
YES
306
EXTRACT LSB OF 63RD COEFFICIENT OF Bi AS THE NEXT BIT OF W
307
SET LSB OF 63RD COEFFICIENT OF Bi TO ZERO
308
UPDATE THE HASH VALUE H BY APPLYING THE HASH FUNCTION TO ITS PREVIOUS VALUE AND ALL THE COEFFICIENTS IN Bi
309
APPLY THE DIGITAL SIGNATURE VERIFICATION ALGORITHM V ON W, H AND THE PUBLIC KEY, Kp: V (W, H, Kp)
310
IS THE RESULT OF V (W, H, Kp) "SUCCESS" ?
NO
YES
311
OUTPUT: "DOES NOT HAVE WATERMARK"
312
OUTPUT: "HAS WATERMARK"
STOP

FIG._5

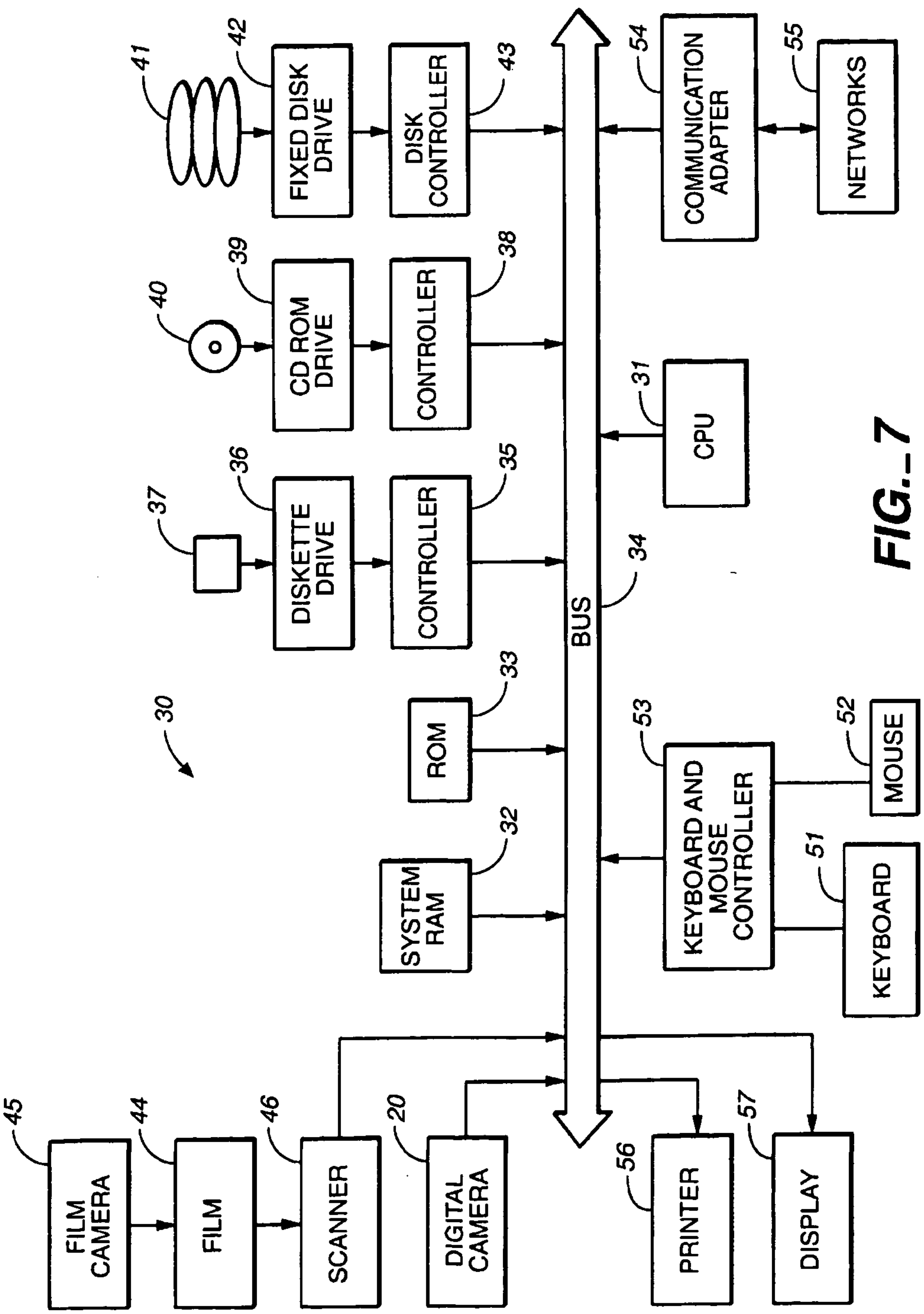

30
41
42
FIXED DISK DRIVE
DISK CONTROLLER
43
40
39
CD ROM DRIVE
CONTROLLER
38
37
36
DISKETTE DRIVE
CONTROLLER
35
ROM
33
SYSTEM RAM
32
BUS
34
54
COMMUNICATION ADAPTER
55
NETWORKS
31
CPU
53
KEYBOARD AND MOUSE CONTROLLER
52
MOUSE
51
KEYBOARD
45
FILM CAMERA
44
FILM
46
SCANNER
20
DIGITAL CAMERA
56
PRINTER
57
DISPLAY

FIG._7

FRAGILE WATERMARKS FOR DETECTING TAMPERING IN IMAGES

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to a fragile watermarking scheme, and more particularly to techniques for inserting and extracting fragile watermarks directly in the frequency domain of a compressed image and for determining whether an image so watermarked has been tampered with.

2. Description of the Related Art

A watermark is a digital pattern inserted into a digital creation, such as a digital image. The process of inserting a watermark into a digital image can be done directly in the frequency domain representation typically used in compressed images. The watermark can be inserted by altering certain frequency coefficients while minimizing the resulting distortion. In case of a block-based frequency domain representation, such as that used in the JPEG image compression standard, frequencies as well as blocks can be chosen to be altered so that the distortion is minimized. In either case, after the watermark has been inserted into the image, a procedure, which may be implemented using certain software, is typically required to validate the watermark.

There are different types of watermarks which serve different purposes. Tamper-resistant watermarks, for example, are designed to identify ownership or the intended recipient of a digital image. To function effectively as an identifier (i.e., to trace unauthorized distribution of an image), a tamper-resistant watermark must be embedded in the image so that it is impossible, or at least difficult, to remove the watermark without visibly damaging the image. Such a watermark must also be resistant to image processing techniques, such as cropping, scaling, image enhancement, compression/decompression, etc. In addition, a tamper-resistant watermark should be readily detectable and recoverable by the proper authorities to permit the tracing and identification of the image, even if someone has tampered with the image.

Another type of watermarks, sometimes referred to as fragile watermarks, are designed to detect tampering of an image. A fragile watermark is embedded in a digital image so that, if someone tampers with the image, that tampering will modify (or destroy) the watermark. Fragile watermarks may be used, for example, in connection with images generated by digital cameras to provide a basis for determining whether or not an image has been tampered with after its creation.

Various watermarking methods have been proposed. For example, U.S. Pat. No. 5,530,759 proposes a digital watermark applied to an original image as a multiplicative correction to pixel sample values of the original image in a linear color space such that the chromaticities of the pixels are not changed. This procedure results in a visible watermark which is simply added to the original image.

U.S. Pat. No. 5,606,609 sets forth an electronic document verification system and method. This refers to a scheme for electronically signing documents, but the signature is not embedded in the document data itself (i.e., the document data is not modified). The signature is just added to the document as another field.

U.S. Pat. Nos. 5,613,004 and 5,687,236 combine steganography (hiding information that is otherwise in plain view) and cryptography (scrambling information that may

be sent by unsecured means). Data is watermarked so that if it is copied, it is possible to determine who owns the original. Thus, the watermarking schemes of these patents are of the tamper-resistant type.

U.S. Pat. No. 5,664,018 proposes a watermarking procedure wherein each of a set of copies of a digitized work has a slightly-modified form of a "baseline" watermark that is placed within a critical region of the data. This is a tamper-resistant scheme that makes it difficult to remove the watermark without damaging the content, even if several parties with differently watermarked copies of a single image collude together in an attempt to remove the watermark.

U.S. Pat. No. 5,689,587 sets forth a method and apparatus for hiding data in images. This is a way to hide information in images, again to ensure tamper resistance (copyright type protection).

However, none of these patents provide a fragile watermarking scheme for detecting tampering. Moreover, none of these patents provide a watermarking technique which may be implemented directly in the frequency domain representation of an image such that the distortion resulting from the embedded watermark is minimized. Yet another shortcoming of these patents is that they do not provide insertion and verification procedures that work without having to completely decompress a compressed image.

OBJECTS OF THE INVENTION

Therefore, it is an object of the present invention to overcome the aforementioned problems.

It is another object of the present invention to provide a scheme for inserting and extracting fragile watermarks in frequency domain and for verifying whether an image so watermarked has been tampered with.

It is still another object of the invention to provide a verification procedure in connection with a fragile watermarking scheme for images, whereby, if tampering has occurred, the verification procedure will reveal a damaged watermark and hence provide a basis for declaring that the image has been tampered with.

It is yet another object of the invention to provide a frequency domain watermark insertion and verification process that uses a secret key only in the insertion step while the verification is carried out using a publicly available key.

It is yet another object of the invention to provide a watermark insertion and verification process that does not require full decompression of a compressed image.

SUMMARY OF THE INVENTION

According to one aspect of the invention, a technique for embedding a fragile watermark in a digital image and a technique for detecting tampering of a digital image so watermarked are provided.

Another aspect of the invention involves watermarking a compressed digital image and detecting tampering of a compressed image so watermarked, without having to completely decompress the image.

Yet another aspect of the invention involves an insertion mechanism that uses a secret key while the corresponding verification mechanism uses a publicly available key.

Watermarking the compressed image initially involves computing a hash value for the image which is accomplished by: partially decoding the compressed digital image to generate a plurality of blocks, each block having a plurality of transform coefficients; obtaining a quantizer of the high-

est frequency coefficient in each block and multiplying that coefficient by its quantizer; determining whether to embed a watermark bit in each block based on the highest frequency transform coefficient in that block and the number of watermark bits remaining to be embedded in the digital image; setting to zero the least significant bit (LSB) of the highest frequency transform coefficient in each block for which it was determined to embed a watermark bit; updating a hash value at each block by applying a hash function at each block, wherein the hash value computed at the last block is a multiple-bit value representative of the entire digital image. Once the hash value for the entire image is computed, a watermark is computed from the computed hash value using a secret key and a digital signature algorithm. Then, each watermark bit is embedded in one of the blocks for which it was previously determined to do so by setting the LSB of the highest frequency transform coefficient in that block to match the corresponding watermark bit.

To determine whether a compressed image so watermarked has been tampered with, the following steps are performed: partially decoding the compressed watermarked image to generate a plurality of blocks, each block having a plurality of transform coefficients; determining each block in which a watermark bit is embedded; extracting, from each block which was previously determined to be an embedder of a watermark bit, the LSB of the highest frequency transform coefficient in that block to generate an extracted watermark; computing a hash value of the digital image by applying a hash function at each block based on the zeroed value of the LSB of the highest frequency transform coefficient in that block, wherein the hash value computed at the last block is a multiple-bit value representative of the entire digital image; applying a digital signature algorithm to the extracted watermark and the computed hash value and a public key to determine whether the compressed watermarked image has been tampered with.

The watermark insertion procedures can be done directly in an image capturing device, such as a digital camera, or can be performed by an appropriately configured computer. Such a computer can also be used to check a watermarked image to determine if tampering has occurred and, if tampering has occurred, to determine where it occurred.

Other objects and attainments together with a fuller understanding of the invention will become apparent and appreciated by referring to the following description and claims taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings wherein like reference symbols refer to like parts:

FIG. **1** is a schematic diagram generally showing the insertion of a watermark into an image in frequency domain in accordance with the invention.

FIG. **2** is a flow diagram depicting an initial scan procedure used in connection with a method of inserting a watermark into an image in frequency domain in accordance with the invention.

FIG. **3** is a flow diagram depicting a procedure to embed bits into an image in connection with a method of inserting a watermark into an image in frequency domain in accordance with the invention.

FIG. **4** is a schematic diagram showing the verification of the existence of the fragile watermark in frequency domain from a watermarked image in accordance with the invention.

FIG. **5** is a flow diagram depicting a method of extracting the watermark from an image and verifying its validity to determine if the image has been tampered with.

FIG. **6** is a block diagram of a digital camera adapted for use in connection with the invention.

FIG. **7** is a block diagram that illustrates the interrelationship between various components that may be used in capturing and viewing digital images, as well as processing such images in accordance with the invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

To watermark an image in the frequency domain, the image is scanned once to compute H, which represents a k-bit hash value of the image. Using a digital signature algorithm, S, and a secret key, K_s, the m-bit watermark $W=S(H,K_s)$ is computed. A second pass of the image is then made to embed the watermark W into the image. The process is illustrated in FIGS. **1**, **2** and **3**.

Referring primarily to FIG. **2** and secondarily to FIGS. **1** and **3**, in the initial scan procedure, the hash value H is initialized to a fixed value in step **101**. The compressed image data, which may be in the form of a JPEG image I, is supplied to a decoder in step **102**. The decoder parses the headers from the JPEG data, noting the value of q, the quantization table entry for the highest frequency coefficient, which is the 63^{rd} coefficient for the 8×8 discrete cosine transform (DCT) used in JPEG. If there is another block of coefficients yet to be decoded and processed (step **103**), the next such block, B_i, is partially decoded in step **104**. Only the entropy coding of the compressed data is undone, avoiding the de-zig-zagging, dequantization, and inverse discrete cosine transform (IDCT) steps needed for full decompression. This results in a representation of B_i made up of only the non-zero quantized coefficients along with their locations in the zig-zag order. Since we use the 63^{rd} coefficient in special ways, the representation of Bi is always made to have the 63^{rd} coefficient even when its value is zero. Note that this can be easily done without de-zig-zagging the other non-zero quantized coefficients because the 63^{rd} coefficient is the last coefficient in the zig-zag order. The decoded representation of the block B_i is then passed to step **105**, where the 63^{rd} coefficient is multiplied by its quantizer, q (obtained in step **102**). The reason for this step is as follows. Small variations in the higher frequencies are invisible to the human eye. The watermark W is embedded bit-by-bit in the image by modifying only the highest frequency coefficient (the 63^{rd} coefficient in this case) so as to minimize the distortion. A watermark bit is embedded (later, in step **207**, FIG. **3**) in a coefficient value by changing the LSB of the value to be equal to the watermark bit. If a quantized 63^{rd} coefficient has the value v, then changing its LSB changes the dequantized coefficient by plus or minus q, where q is the quantizer for the 63^{rd} coefficient. To minimize this change, we set the quantizer for the 63^{rd} coefficient to be 1 and replace the 63^{rd} coefficient in every block (not just those embedding watermark bits) directly by their dequantized values (which are obtained by multiplying by q). On decompression, the distortion can only be plus or minus 1, as a result of this step. Since the 63^{rd} coefficient is typically zero in most of the blocks, the increase in compressed size resulting from the multiplication in step **105** is minimal, as only the non-zero coefficients account for most of the compressed size.

A decision is made as to whether a watermark bit is to be embedded in block B_i in step **106**. The decision procedure for step **106** is designed so as to minimize the resulting distortion in the image as well as to minimize the resulting size-increase of the compressed image. We denote this

decision procedure by the name EMBEDDER-TEST, to simplify the subsequent presentation as this decision procedure is used again in two places. EMBEDDER-TEST is fully described as follows.

For color images, watermark bits are embedded only in the luminance plane of the image. This is done so that during decompression, when the luminance-chrominances color representation is converted back to red, green, and blue pixel values (RGB), the resulting distortion is minimized. Moreover, the chrominance planes are typically subsampled, so any distortion in a single chrominance block results in distortions in several RGB blocks. Thus, in grayscale images as well as in color images, watermark bits are embedded only in the color component numbered zero (which is the luminance plane for color images). To minimize the distortion, watermark bits are embedded only in the 63^{rd} DCT coefficient, as mentioned previously. To minimize the compressed size, only those blocks are chosen to embed a watermark bit where the 63rd coefficient is already non-zero. This follows from the observation that changing a zero value to a non-zero value results in a far greater increase in compressed size, compared to changing a non-zero value to another nonzero value. However, since EMBEDDER-TEST will also be performed by the watermark verification procedure, we do not choose as embedders even the blocks where the 63^{rd} coefficient (dequantized) is plus or minus 1, as it might potentially be turned to zero on embedding the watermark bit, and then the verifier will not be able to decide the block to be an embedder. If, at some point, the number of watermark bits remaining to be embedded becomes equal to the number of blocks remaining in component zero, every subsequent block in component zero is decided upon as an embedder of a watermark bit.

If the decision to embed a bit in block B_i is yes, then the LSB of the 63^{rd} DCT coefficient is set to zero in step **107** and the procedure proceeds to step **108**. If the decision is no, then the procedure directly proceeds to step **108**. The hash value H is updated in step **108** using its previous value, and the values of all the non-zero quantized coefficients in B_i along with their locations in the zig-zag order, using a one-way hash function. The 63^{rd} coefficient is always included in this computation even if it is zero.

When all the blocks have been processed, the procedure proceeds to step **109**, where the digital signature algorithm S is applied to the computed hash value, H, and the secret key, K_s, to compute the m-bit watermark $W=S(H, K_s)$.

Referring primarily to FIG. **3** and secondarily to FIGS. **1** and **2**, in the second pass of the watermark insertion procedure, the input JPEG image I is once again fed to a decoder which parses its headers noting the value of q, the quantizer for the 63^{rd} coefficient, in step **201**. The headers for the watermarked output JPEG image, I_w, are produced in step **202** by copying them directly from the input image, except that the quantizer of the 63^{rd} coefficient in the quantizer table is changed to be 1 instead of its previous value, q. If there is another block of coefficients yet to be decoded and processed (step **203**), the next such block, B_i, is partially decoded in step **204**. Again, only the entropy coding of the compressed data is undone, avoiding the de-zig-zagging, dequantization, and IDCT steps needed for full decompression. This results in a representation of B_i made up of only the non-zero quantized coefficients (except for the 63^{rd} coefficient which is always included in the representation) along with their locations in the zig-zag order. The 63^{rd} coefficient of each block is multiplied by the q, in step **205**. EMBEDDER-TEST is performed in step **206** to determine whether block B_i is supposed to embed the next bit of W. This determination may be made again on a block-by-block basis or may be made using the results of the previous decision-making process (step **105**), if those results are stored in memory. In any case, if block B_i is supposed to embed the next bit of W, then the LSB of the 63^{rd} DCT coefficient of B_i is set to match the next bit of W in step **207** and the procedure proceeds to step **208**. If the decision in step **206** is "no", then the procedure directly proceeds to step **208**. In step **208**, the coefficients in B_i are encoded and produced as output into the compressed data stream for the watermarked image, I_w. Note that the representation of the quantized coefficients of B_i that we use enables efficient encoding as the quantized coefficients are already in the zig-zag order, thus avoiding the DCT, quantization, and zig-zagging steps generally required for compression. The process repeats until all of the blocks have been processed.

The verification procedure for the frequency domain watermarking method is illustrated in FIGS. **4** and **5** and is used to determine if the image has been tampered with. Referring to FIG. **5**, after initializing the hash value H in step **301**, the watermark verification procedure begins to decode the input image, I_w, parsing its headers, in step **302**. In step **303**, it is determined whether another block remains to be decoded. If so, the next block, B_i, is partially decoded in step **304**. Again, only the entropy coding of the compressed data is undone, avoiding the de-zig-zagging, dequantization, and IDCT steps needed for full decompression. This results in a representation of B_i made up of only the non-zero quantized coefficients (except for the 63^{rd} coefficient which is always included in the representation) along with their locations in the zig-zag order. EMBEDDER-TEST is performed in step **305** to determine whether block B_i is supposed to embed the next bit of W. If it is, then the LSB of the 63^{rd} coefficient of B_i is extracted as the next bit of the watermark W in step **306**, and then that LSB is set to zero in step **307**. The procedure moves to step **308** from step **307** as well as from step **305** when the block B_i is not a watermark bit embedder. The hash value H is updated in step **308** using its previous value, and the values of all the non-zero quantized coefficients in B_i along with their locations in the zig-zag order, using the one-way hash function. The 63^{rd} coefficient is always included in this computation even if it is zero. The process continues through all the blocks and in the end, the extracted watermark W and the hash value H have been fully computed. At this point the digital signature verification algorithm V (corresponding to the signing algorithm S) is applied using the public key K_p (corresponding to the secret key K_s) to verify whether or not W is the same as $S(H, K_s)$ in step **309**. Note that step **309** does not require the use of the secret key K_s. Step **310** examines the output of the digital signature verification algorithm $V(W,H,K_p)$ applied in step **309**. If the verification algorithm $V(W,H,K_p)$ is successful, then the image has the fragile watermark intact, and thus has not been tampered with (step **312**). If $V(W,H,K_p)$ outputs failure, then the fragile watermark (if it ever existed) has been destroyed, and it may be concluded that the image has been tampered with or never had the fragile watermark corresponding to the key pair (K_s,K_p) (step **311**).

The effectiveness of the image tampering detection process depends on the strength of the hash function and the signing and verification algorithms S and V. A variety of one-way hash functions may be used, such as a hash function called MD5 developed by R. Rivest, or the SHA, or RIPEMD hash functions. Similarly, there is considerable choice available for the signature and verification algorithm pair, such as the El Gamal scheme, the DSA algorithm, or the RSA algorithm.

This fragile watermarking procedure can be modified to detect roughly the area of the image in which some tampering has been done. This is accomplished by dividing the image into some number of regions and applying the whole insertion procedure separately on each region. Only the regions that contain tampering will show a damaged watermark. This modification will not, however, detect a restricted form of tampering which is done by forming an image that is a collage of several regions extracted from different images or reordered regions from a single image, each region carrying a valid watermark. Such tampering, however, is likely to be visually obvious, if the regions are large enough.

Similar fragile watermarking techniques can be applied in the spatial domain as well. Instead of the highest-frequency coefficients, all or some of the pixels can be directly used as watermark bit embedders by setting their LSB to zero prior to the hash computation and then setting that LSB to the watermark bit in the second pass. Note that this invention does not suggest intermingling of the spatial and frequency domain watermarking processes. It does not suggest that watermark insertion can be done in the frequency domain as revealed in this invention while the corresponding watermark verification can be done in the spatial-domain as revealed in this invention.

In addition, the watermark can be chosen to be a visible watermark. A visual watermark can be embedded in the frequency domain by computing the transform frequency coefficients of a distinctive watermarking signal, and simply adding the coefficients to those of the image. This process works as a result of linearity of the transforms commonly used in image coding (such as DCT), which ensures that addition in the pixel domain corresponds to addition in the frequency domain. A similar visible watermarking process can be performed in spatial-domain as well.

It should be noted that the block and flow diagrams used to illustrate the watermark insertion, extraction and verification procedures of the present invention, illustrate the performance of certain specified functions and relationships thereof. The boundaries of these functional blocks have been arbitrarily defined herein for the convenience of description. Alternate boundaries may be defined so long as the specified functions and relationships thereof are appropriately formed. Moreover, the flow diagrams do not depict syntax or any particular programming language. Rather, they illustrate the functional information one skilled in the art would require to fabricate circuits or to generate software to perform the processing required. Each of the functions depicted in the block and flow diagrams may be implemented, for example, by software instructions, a functionally equivalent circuit such as a digital signal processor circuit, an application specific integrated circuit (ASIC) or combination thereof.

The watermarking techniques of the present invention may be employed in connection with various devices including a digital camera, a block diagram of which is illustrated in FIG. 6. Operating under microprocessor control, the digital camera **20** has a charge-coupled device (CCD) image sensor that captures an image and converts it to an analog electrical signal in block **21**. The analog signal is then processed and digitized in block **22**, after which the digital image is temporarily stored in a frame buffer **23** while it undergoes digital processing in block **24**. The digital image processing block **24** performs several functions including compression and decompression. Processing block **24** may also perform the watermarking techniques of the present invention using hardware or software. Under user control **25**, the processing block **24** interfaces with in-camera image storage **26** where decompressed image data may be stored. The storage block **26** may comprise compact magnetic or solid-state storage media, either removable or fixed within the camera **20**, and may include removable, large-capacity PCMCIA-format hard disk cards or flash memory cards.

The camera **20** includes analog and digital outputs, **27** and **28** respectively, through which image data may be transmitted within the camera or to external devices. Uncompressed image data may be transmitted, via the analog outputs **27**, to an LCD screen **29** within the camera **20**, or to external devices such as a VCR or television receiver. Image data, whether compressed or uncompressed, may also be transmitted through the digital outputs **29** to a digital device such as a computer system where the image could be displayed or where watermarked images could be verified.

FIG. 7 is a block diagram that illustrates the interrelationship between various components that may be used in capturing, processing and viewing digital images. One of the more important components is a computer system, identified generally by reference numeral **30**. The computer system **30** may be of any suitable type such as a main frame or personal computer.

Computer system **30** comprises a central processing unit (CPU) **31** which may be a conventional microprocessor, a random access memory (RAM) **32** for temporary storage of information, and a read only memory (ROM) **33** for permanent storage of information. Each of these components is coupled to a bus **34**. Operation of the computer system **30** is typically controlled and coordinated by operating system software. The operating system, which is embodied in the system memory and runs on CPU **31**, coordinates the operation of computer system **30** by controlling allocation of system resources and performing a variety of tasks, such as processing, memory management, networking and I/O functions, among others.

Also coupled to bus **34** by a controller **35** is a diskette drive **36** into which a non-volatile mass storage device such as a diskette **37** may be inserted. Similarly, a controller **38** interfaces between bus **34** and a compact disc (CD) ROM drive **39** which is adapted to receive a CD ROM **40**. A hard disk **41** is provided as part of a fixed disk drive **42** which is coupled to bus **34** by a disk controller **43**.

Software for the watermarking techniques may be stored on storage devices **207** and **210** and transferred to CPU **31** for execution. Alternatively, the software may be stored in RAM **32** or ROM **33**. Similarly, image data be loaded into and extracted from computer system **30** using removable storage media devices such as the diskette **37** and CD ROM **40**.

Image data may be input into computer system **30** in other ways as well. Film-based images **44** generated by a film camera **45** can be digitized by a scanner **46** for storage and processing by the computer **30**. The digital camera **20** can directly digitize images and transmit them to the computer **30**, as explained above. A keyboard **51** and mouse **52**, which are coupled to bus **34** via a controller **53**, facilitate the input of such data and otherwise provide a means for entering information into computer system **30**.

Image data may also be transferred to and from computer **30** for remote locations. To this end, computer **30** may also include a communications adapter **54** which enables the computer **30** to communicate with networks **55**, which may include local area networks (LANs), the internet or online services, via direct connections or via modem.

In accordance with the invention, images that have been previously watermarked, say, in the digital camera **20** may

be transmitted to computer **30** for verification. Alternatively, unmarked images may be watermarked and later verified in computer **30** using appropriate hardware or software that is executed by the CPU **31**.

Digital images transmitted or stored in computer **30** may be viewed in a number of different ways. A printer **56** attached to computer **30** can produce color prints that vary in quality depending on the printer **56**. Another option is to view the images on a display **57** associated with the computer **30**. Yet another choice is to display the images on a television monitor using a VCR.

While the invention has been described in conjunction with specific embodiments, it will be evident to those skilled in the art in light of the foregoing description that many further alternatives, modifications and variations are possible. For example, visible watermarking process and the invisible watermarking process described in this invention can be combined depending upon the application. Furthermore, the choice of blocks to be used in the watermarking process can be made application dependent. Thus, the invention described herein is intended to embrace all such alternatives, modifications, applications and variations as may fall within the spirit and scope of the appended claims.

What is claimed is:

1. A method of watermarking a compressed digital image, comprising the steps of:

partially decoding the compressed digital image to generate a plurality of data blocks, each block having a plurality of transform coefficients;

determining whether to embed a watermark bit in each block based on one of the plurality of transform coefficients in that block and a number of watermark bits remaining to be embedded in the digital image;

computing a watermark, having a plurality of bits, for the entire digital image by applying a digital signature algorithm and a secret key; and

embedding a watermark bit in each block for which it was determined to do so in said determining step by setting the one of the plurality of transform coefficients in that block to match a corresponding bit of the computed watermark.

2. The method of claim **1**, wherein the one of the plurality of transform coefficients in each block is the transform coefficient representative of the highest vertical frequency and highest horizontal frequency in that block.

3. The method of claim **1**, wherein said step of computing the watermark further comprises the step of:

setting to zero at least one bit of the one of the plurality of transform coefficients in each block for which it is determined in said determining step to embed a watermark bit.

4. The method of claim **3**, further comprising the step of:

encoding the plurality of blocks after said embedding step is completed to re-compress the watermarked image.

5. The method of claim **3**, wherein said of computing the watermark further comprises the step of:

updating a hash value at each block by applying a hash function at each block, wherein the hash value computed at a last block is the multiple-bit hash value representative of the entire digital image.

6. The method of claim **5**, wherein said step of computing the watermark further comprises the steps of:

obtaining a quantizer of the one of the plurality of transform coefficients in each block; and

multiplying the one of the plurality of transform coefficients in each block by its quantizer.

7. A method of watermarking a compressed digital image, comprising the steps of:

computing a watermark, having a plurality of bits, for the entire digital image, comprising the steps of:

partially decoding the compressed digital image to generate a plurality of data blocks, each block having a plurality of transform coefficients;

obtaining a quantizer of the one of the plurality of transform coefficients in each block;

multiplying the one of the plurality of transform coefficients in each block by its quantizer;

determining whether to embed a watermark bit in each block based on one of the plurality of transform coefficients in that block that is representative of a highest vertical frequency and a highest horizontal frequency in that block and a number of watermark bits remaining to be embedded in the digital image;

setting to zero at least one bit of the one of the plurality of transform coefficients in each block for which it is determined in said determining step to embed a watermark bit;

updating a hash value at each block by applying a hash function at each block, wherein the hash value computed at a last block is a multiple-bit hash value representative of the entire digital image; and

applying a secret key and a digital signature algorithm to the multiple-bit hash value to compute the watermark; and

embedding a watermark bit in each block for which it was determined to do so in said determining step by setting the at least one bit of the one of the plurality of transform coefficients in that block to match a corresponding bit of the computed watermark.

8. A method of detecting tampering of a compressed watermarked image, comprising the steps of:

partially decoding the compressed watermarked image to generate a plurality of data blocks, each block having a plurality of transform coefficients;

determining each block in which a watermark bit is embedded;

extracting, from each block for which it was determined in said determining step to be an embedder of a watermark bit, at least one bit of one of the plurality of transform coefficients in that block to generate an extracted watermark;

computing a hash value of the digital image by applying a hash function at each block based on a zeroed value of the at least one bit of one of the plurality of transform coefficients in that block, wherein the hash value computed at a last block is a multiple-bit value representative of the entire digital image; and

applying a digital signature verification algorithm to the extracted watermark and the multiple-bit computed hash value and a public key to determine whether the compressed watermarked image has been tampered with.

9. The method of claim **8**, wherein the one of the plurality of transform coefficients in each block is the transform coefficient representative of the highest vertical frequency and highest horizontal frequency in that block.

10. An image capturing device, comprising:

a sensor for capturing light and converting the light into an analog image signal;

an analog-to-digital converter for converting the analog image signal to a digital image; and

a digital image processor for compressing the digital image to generate a plurality of data blocks, each block

having a plurality of transform coefficients, determining whether to embed a watermark bit in each block based on one of the plurality of transform coefficients in that block and a number of watermark bits remaining to be embedded in the digital image, computing a watermark, having a plurality of bits, for the entire digital image, and embedding a watermark bit in each block for which it was determined to do so by setting the one of the plurality of transform coefficients in that block to match a corresponding bit of the computed watermark.

11. The image capturing device of claim **10**, wherein said digital image processor sets to zero at least one bit of the one of the plurality of transform coefficients in each block for which it is determined to embed a watermark bit.

12. The image capturing device of claim **11**, wherein said digital image processor encodes the plurality of blocks after the watermark bit embedding is completed to re-compress the watermarked image.

13. The image capturing device of claim **11**, wherein said digital image processor updates a hash value at each block by applying a hash function at each block, wherein the hash value computed at a last block is the multiple-bit hash value representative of the entire digital image.

14. The image capturing device of claim **11**, wherein said digital image processor obtains a quantizer of the one of the plurality of transform coefficients in each block and multiples the one of the plurality of transform coefficients in each block by its quantizer.

15. A computer system including a processor and a memory having a computer-readable readable program code embodied therein for causing the processor to detect tampering of a compressed watermarked digital image by performing the steps of:

partially decoding the compressed watermarked image to generate a plurality of data blocks, each block having a plurality of transform coefficients;

determining each block in which a watermark bit is embedded;

extracting, from each block for which it was determined in said determining step to be an embedder of a watermark bit, at least one bit of one of the plurality of transform coefficients in that block to generate an extracted watermark;

computing a hash value of the digital image by applying a hash function at each block based on a zeroed value of the at least one bit of one of the plurality of transform coefficients in that block, wherein the hash value computed at a last block is a multiple-bit value representative of the entire digital image; and

comparing the extracted watermark with the multiple-bit value by applying a digital signature verification algorithm to determine whether the compressed watermarked image has been tampered with.

16. The computer system of claim **15**, wherein the one of the plurality of transform coefficients in each block is the transform coefficient representative of the highest vertical frequency and highest horizontal frequency in that block.

17. A method of watermarking a digital image, having a plurality of pixels, comprising the steps of:

computing a watermark, having a plurality of bits, for the entire digital image, comprising the steps of:

determining whether to embed a watermark bit in each pixel based on one of a plurality of bits representative of that pixel and a number of watermark bits remaining to be embedded in the digital image;

setting to zero at least one bit of each pixel for which it is determined in said determining step to embed a watermark bit;

updating a hash value at each pixel by applying a hash function at each pixel, wherein the hash value computed at a last pixel is a multiple-bit hash value representative of the entire digital image; and

applying a secret key and a digital signature algorithm to the multiple-bit hash value to compute the watermark; and

embedding a watermark bit in each pixel for which it was determined to do so in said determining step by setting the at least one bit of that pixel to match a corresponding bit of the computed watermark.

18. The method of claim **17**, wherein the at least one bit is the least significant bit.

19. A method of detecting tampering of a watermarked digital image, having a lo plurality of pixels, comprising the steps of:

determining each pixel in which a watermark bit is embedded;

extracting, from each pixel for which it was determined in said determining step to be an embedder of a watermark bit, at least one of a plurality of bits representative of that pixel to generate an extracted watermark;

computing a hash value of the digital image by applying a hash function at each pixel based on a zeroed value of the at least one bit of that pixel, wherein the hash value computed at a last pixel is a multiple-bit value representative of the entire digital image; and

applying a digital signature verification algorithm to the extracted watermark and the multiple-bit computed hash value and a public key to determine whether the compressed watermarked image has been tampered with.

20. The method of claim **19**, wherein the at least one bit is the least significant bit.

21. A method of watermarking a compressed digital image, comprising the steps of:

adding a visible watermark;

computing an invisible watermark from the digital image and the visible watermark, comprising the steps of:

partially decoding the compressed digital image to generate a plurality of data blocks, each block having a plurality of transform coefficients;

obtaining a quantizer of the one of the plurality of transform coefficients in each block;

multiplying the one of the plurality of transform coefficients in each block by its quantizer;

determining whether to embed an invisible watermark bit in each block based on one of the plurality of transform coefficients in that block that is representative of a highest vertical frequency and a highest horizontal frequency in that block and a number of invisible watermark bits remaining to be embedded in the digital image;

setting to zero at least one bit of the one of the plurality of transform coefficients in each block for which it is determined in said determining step to embed an invisible watermark bit;

updating a hash value at each block by applying a hash function at each block, wherein the hash value computed at a last block is a multiple-bit hash value representative of the entire digital image; and

applying a secret key and a digital signature algorithm to the multiple-bit hash value to compute the invisible watermark; and

embedding an invisible watermark bit in each block for which it was determined to do so in said determining step by setting the at least one bit of the one of the plurality of transform coefficients in that block to match a corresponding bit of the computed invisible watermark.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,064,764
DATED : May 16, 2000
INVENTOR(S) : Vasudev Bhaskaran, et al.

It is certified that errors appear in the above identified patent and that said Letters Patent is hereby corrected as shown below:

Title page, item 56, References Cited, US Patent Documents, change "5,942,414" to --5,946,414--.

Column 12, line 13, delete "lo".

Signed and Sealed this

Twenty-fourth Day of April, 2001

Attest:

Nicholas P. Godici

NICHOLAS P. GODICI

Attesting Officer *Acting Director of the United States Patent and Trademark Office*